

Procesamiento de imágenes médicas en odontología como ayuda al diagnóstico

**Jorge Sánchez-Barriga Pons
Marcos Rebollo Fernández**

Madrid, junio de 2017

**Proyecto de la Facultad de Informática
Universidad Complutense de Madrid**



**Trabajo Fin de Grado en Ingeniería de Computadores y Software
Curso 2016-2017**

Director: Gonzalo Pajares Martinsanz
Codirector: Vicente Vera González

Agradecimientos Personales

Jorge Sánchez-Barriga Pons

A mis padres, hermana y abuelos, gracias por el inmenso sacrificio, apoyo, comprensión, cariño, y todo lo que aportáis hacia mi persona, sin vosotros nada de esto sería posible.

Mi compañero y amigo Marcos, gracias por haber compartido conmigo todos estos años de carrera y tantos proyectos, para finalmente, poder realizar este TFG juntos.

Gracias a Gonzalo Pajares por ayudarnos y apoyarnos en el proyecto.

Marcos Rebollo Fernández

Gracias a mis padres, Carlos y Amelia, por haberme apoyado siempre, por el inmenso sacrificio realizado durante todos estos años, por el cariño y vuestra infinita paciencia, un gracias se queda corto por todo lo que habéis hecho.

Gracias a mi hermano Álvaro, has sido un modelo a seguir durante todos estos años, no podría haber tenido un hermano mejor.

A mis amigos y compañeros de facultad, con los que tantas horas de sufrimiento y alegrías he compartido durante todo este tiempo, especialmente a Jorge, ha sido un placer haber hecho este TFG contigo.

Gracias a todo aquellos que habéis estado ahí, especialmente en los malos momentos, una vez más gracias,

Agradecimientos

A nuestro director, Dr. Gonzalo Pajares, por dejarnos la libertad de elegir por nosotros mismos el proyecto que queríamos realizar, y depositar la confianza necesaria en nosotros para lograrlo.

Al Dr. Vicente Vera por las valiosas indicaciones técnicas para desarrollar el trabajo y por el abundante material proporcionado en forma de imágenes para poder llevarlo a buen puerto.

Índice

Listado de abreviaturas.	5
Listado de figuras.	6
Resumen / Abstract.	7
1. Introducción.	9
1.1. Consideraciones preliminares.	9
1.2. Motivación, objetivos y plan de trabajo.	9
1.3. Estructura de la memoria.	11
1.4. Introduction	11
1.5. Contribuciones personales.	13
2. Funcionalidades de la aplicación.	17
2.1. Cálculo del índice de Taguchi.	18
2.1.1. Generalidades sobre el Índice de Taguchi.	18
2.1.2. Consideraciones sobre el cálculo del Índice de Taguchi.	19
2.2. Descripción de Texturas.	19
2.2.1. Generalidades sobre el cálculo de texturas.	19
2.2.2. Consideraciones sobre el cálculo de texturas.	20
3. Desarrollo.	20
3.1. Pre-procesamiento.	21
3.2. Cálculo del índice de Taguchi.	21
3.2.1. Detección automática del borde.	21
3.2.1.1. Recorte de la imagen.	22
3.2.1.2. Binarización de la imagen.	22
3.2.1.3. Limpieza de ruido.	23
3.2.1.4. Detección de bordes.	23
3.2.2. Generación de la primera recta.	24
3.2.2.1. Transformada de Hough.	24
3.2.3. Generación de la recta paralela.	25
3.2.3.1. Cálculo de la pendiente.	26
3.2.3.2. Cálculo de coordenadas.	26
3.2.4. Generación de la recta perpendicular.	26
3.2.4.1. Cálculo de la pendiente.	26
3.2.4.2. Cálculo de coordenadas.	27

3.2.5.	Intersecciones entre rectas.	27
3.2.5.1.	Punto de intersección entre rectas.	27
3.2.6.	Distancia entre los puntos generados.	28
3.3.	Descripción de texturas.	28
3.3.1.	Pre-procesamiento.	28
3.3.2.	Cálculo de la desviación típica.	28
4.	Diseño.	29
5.	Implementación y resultados.	31
5.1.	Pre-procesamiento.	31
5.2.	Cálculo del índice de Taguchi.	32
5.2.1.	Cálculo automático del borde inferior.	32
5.2.1.1.	Recorte de la imagen.	32
5.2.1.2.	Binarización de la imagen.	33
5.2.1.3.	Limpieza de ruido.	34
5.2.1.4.	Detección de bordes.	34
5.2.2.	Generación de recta ajustada al borde inferior.	35
5.2.2.1.	Transformación a uint8	35
5.2.2.2.	Extracción de blancos.	35
5.2.2.3.	Transformada de Hough.	35
5.2.2.4.	Unión de líneas similares.	36
5.2.3.	Generación de la recta paralela.	37
5.2.3.1.	Cálculo de la pendiente.	37
5.2.3.2.	Cálculo de coordenadas.	37
5.2.4.	Generación de la recta perpendicular.	38
5.2.4.1.	Cálculo de la pendiente.	38
5.2.4.2.	Cálculo de coordenadas.	38
5.2.5.	Intersecciones entre rectas.	39
5.2.5.1.	Punto de intersección entre rectas.	39
5.2.5.2.	Distancia entre los puntos generados.	40
5.3.	Descripción de texturas.	40
6.	Conclusiones y trabajo futuro.	41
	Bibliografía.	43
	Anexos: Manual de Usuario y Código.	45
	Anexos A: Manual de usuario.	45
	Anexos B: Código.	52

Listado de abreviaturas

<i>TFG</i>	<i>Trabajo Fin de Grado</i>
<i>UCM</i>	<i>Universidad Complutense de Madrid</i>
<i>ICM</i>	<i>Índice Cortical Mandibular</i>
<i>GUI</i>	<i>Graphic User Interface</i>
<i>DMO</i>	<i>Densidad Media Ósea</i>

Listado de figuras

Figura 1	Diagrama del cálculo del ICM	18
Figura 2	Identificación del agujero mentoniano	27
Figura 3	Diagrama de clases	30
Figura 4	Diagrama de actividad	31
Figura 5 (a)	Imagen original	32
Figura 5 (b)	Imagen transformada a una nueva escala de grises	32
Figura 6 (a)	Selección rectangular de la imagen	33
Figura 6 (b)	Sección ampliada	33
Figura 7 (a)	Imagen original	33
Figura 7 (b)	Imagen binarizada	33
Figura 8 (a)	Imagen binarizada original	34
Figura 8 (b)	Imagen binarizada sin ruido	34
Figura 9	Detección de bordes mediante operador Sobel	34
Figura 10	Recta autogenerada tangencial al borde inferior de la mandíbula	36
Figura 11	Recta paralela tangencial al borde superior de la mandíbula	38
Figura 12	Recta perpendicular trazada desde el agujero mentoniano	39
Figura 13	Intersección entre rectas e ICM calculado	40
Figura 14	Imagen resultante del cálculo de texturas	41
Figura 15	Interfaz de la aplicación al iniciar	45
Figura 16	Botón de la GUI que permite la carga de archivos	46
Figura 17	Carga de archivos en la aplicación	46
Figura 18	Visualización de la radiografía	47
Figura 19	Botón de la GUI que permite el recorte de la imagen	48
Figura 20	Recuadro para seleccionar la zona de recorte	48
Figura 21	Recta auto detectada del borde inferior de la mandíbula	49
Figura 22	Recta paralela a la recta autogenerada	49
Figura 23	Recta perpendicular a ambas rectas y cálculo del ICM	50
Figura 24	Botón de la GUI que permite el cálculo de la textura	51
Figura 25	Gráfico de intensidad de texturas	51

Resumen

En el campo de la odontología es necesario e importante tener una herramienta que permita realizar distintos tipos de cálculos sobre una imagen dada, en este caso, sobre una radiografía mandibular.

Con este fin, hemos aplicado los conocimientos adquiridos en procesamiento de imágenes y programación para crear una aplicación que permita obtener resultados para los objetivos requeridos.

Uno de tales objetivos que a su vez constituye el fundamental de la aplicación es obtener el Índice Cortical Mandibular (ICM), se trata de una medida importante ya que permite detectar signos de osteoporosis a nivel maxilar y mandibular.

Otro objetivo es el estudio sobre el análisis de las texturas óseas sobre radiografías panorámicas de la mandíbula. Esto permite al especialista poder detectar diferentes irregularidades bucales como puede ser una infección en las encías, aislar y realzar la zona afectada o detectar enfermedades en la mandíbula como la osteoporosis. Todo ello como consecuencia del estudio de las diferentes texturas presentes en la imagen bajo análisis.

En este proyecto se han utilizado las técnicas propias de lo que se conoce como percepción computacional desde la perspectiva del procesado de imágenes. Aunque el objetivo principal es una automatización del proceso del cálculo del ICM y la descripción de texturas, en determinados pasos de la ejecución de la aplicación se deja al usuario libertad para escoger los parámetros requeridos por determinadas rutinas implementadas en la aplicación, esto es debido a la dificultad intrínseca de automatizar la localización de determinadas regiones necesarias para calcular el ICM.

Para unir e integrar todas las funcionalidades y conceptos se hace uso de la herramienta Matlab (2017), que a través de la ventana gráfica permite un uso más intuitivo y sencillo de la aplicación.

Palabras clave: procesamiento imágenes, odontología, índice cortical mandibular, descripción de texturas

Abstract

In the field of dentistry it's both important and required to rely on a tool which allows the user to achieve several estimations from an image, particularly, a mandibular radiography.

With this goal in mind, and also using our knowledge in the field of digital image processing, we have implemented an app which accomplish such objectives.

The main aim of the project is to work out the Mandibular Cortical Index (ICM), this index is a significant measure because it allows to detect early osteoporosis symptoms at the level of the maxilla and mandible.

Another objective is the study and analysis of the bone texture using panoramic radiographs, This allows the user to detect several anomalies such as dental abscess or osteoporosis. This is a direct consequence of the study and analysis of the different textures shown in the image.

In this project we have put the techniques learnt in Computational Perception to use, specially digital image processing. Although the main goal of the project is to automate the estimation of the ICM, the user is given some freedom in several steps of the execution of the app, he may be required to choose some parameters needed to make the app work, this happens because some of the steps to achieve the ICM are hard to automate.

In order to unify all the concepts and functionalities of the project we use the multi-paradigm numerical computing environment Matlab (2017), which uses a Graphical User Interface making it easier and intuitive to the user.

Keywords: image processing, odontology, Mandibular Cortical Index, texture description.

1. Introducción

1.1. Consideraciones preliminares

En el campo de la odontología es necesario e importante tener una herramienta que permita realizar distintos tipos de cálculos sobre una imagen dada, en este caso, sobre una radiografía mandibular.

Con este fin, hemos aplicado los conocimientos adquiridos en procesamiento de imágenes y programación para crear una aplicación que permita obtener resultados para los objetivos requeridos.

Uno de tales objetivos, que a su vez constituye el fundamental de la aplicación es obtener el Índice Cortical Mandibular (ICM), se trata de una medida importante ya que permite detectar signos de osteoporosis a nivel maxilar y mandibular.

Otro objetivo es el estudio sobre el análisis de las texturas óseas sobre radiografías panorámicas de la mandíbula. Esto permite al especialista poder detectar diferentes irregularidades bucales como puede ser una infección en las encías, aislar y realzar la zona afectada o detectar enfermedades en la mandíbula como la osteoporosis. Todo ello como consecuencia del estudio de las diferentes texturas presentes en la imagen bajo análisis.

En este proyecto se han utilizado técnicas propias de lo que se conoce como percepción computacional desde la perspectiva del procesado de imágenes. Aunque el objetivo principal es una automatización del proceso del cálculo del ICM y la descripción de texturas, en determinados pasos de la ejecución de la aplicación se deja al usuario libertad para escoger los parámetros requeridos por determinadas rutinas implementadas en la aplicación, esto es debido a la dificultad intrínseca de automatizar la localización de determinadas regiones necesarias para calcular el ICM.

Para unir e integrar todas las funcionalidades y conceptos se hace uso de la herramienta Matlab (2017), que a través de la ventana gráfica permite un uso más intuitivo y sencillo de la aplicación.

1.2. Motivación, objetivos y plan de trabajo

El planteamiento que sustenta el proyecto consiste en automatizar al máximo el proceso de tratamiento de las imágenes radiológicas con el fin de simular un problema real como el que se describe a continuación. Lo más idóneo es que en la propia consulta de odontología se obtenga de manera sencilla y rápida, probablemente en cuestión de segundos, la información necesaria a partir de una radiografía de un paciente, de forma que el odontólogo especialista obtenga la máxima información cuantitativa posible al instante para

emitir un posible diagnóstico, que transmite al momento al paciente o informa a otros profesionales en caso de tener que tomar una decisión conjunta o realizar alguna prueba específica adicional al paciente.

Técnicamente, la visión general del producto consiste en el desarrollo de una aplicación informática que permita la visualización de ciertos datos para un mejor diagnóstico.

Desde el punto de vista médico son dos los objetivos que se plantean en el presente proyecto:

1. Medir el índice de Taguchi (2010) que proporciona un valor para determinar el grado de osteoporosis de un paciente, mediante medidas de distancia (en píxeles) en la base de la mandíbula.
2. Identificar diferentes tipos de texturas con el fin de discernir entre estructuras óseas de otros tejidos y dentro de ellas diferentes niveles de intensidad, que podrían ser indicativos de infecciones u otros desórdenes a nivel bucal.

Desde el punto de vista informático y teniendo en cuenta las consideraciones previas, se plantean los siguientes objetivos, que se transforman posteriormente en requisitos:

1. La interfaz de usuario debe ser simple e intuitiva.
2. Poner énfasis en la visualización de las imágenes radiografías para no perder calidad, ya que es la principal herramienta de trabajo del cliente.
3. Integrar de todos los elementos en la aplicación y que éstos sean fáciles de identificar y usar.
4. Obtener la mayor fiabilidad posible en las medidas realizadas como ayuda al diagnóstico
5. Desarrollar una aplicación modular que permita la incorporación de nuevas funcionalidades sin necesidad de rediseñar la aplicación.

Para cumplir con los objetivos expuestos, se propone el siguiente plan de trabajo, desglosado por objetivos y duración de los mismos teniendo en cuenta la duración aproximada de ocho meses del proyecto, que hace un total de 32 semanas

Objetivo	Duración (semanas)
Desarrollo de la interfaz de usuario	2
Preprocesamiento de la imagen y funcionalidades complementarias	2
Auto detección de borde inferior	10
Trazado de rectas secundarias	7
Descripción de texturas	4
Redacción de la memoria	7 (distribuidas a lo largo del proyecto)

1.3. Estructura de la memoria

La memoria está distribuida en cinco secciones claramente diferenciados en los que se detalla todo el proceso de investigación y desarrollo del proyecto. Además se incluye un apéndice con el fin de matizar y dejar suficientemente claros algunos aspectos y funcionamiento de la aplicación.

La primera sección se compone de una breve introducción sobre la razón de ser de este proyecto, que incluye las dos secciones previas, así como las contribuciones personales de cada miembro del equipo, que se detallan a continuación.

En la segunda sección se describe la funcionalidad general de la aplicación dividiéndolas en dos subapartados que detallan el origen el problema a abordar y el enfoque de la solución que se proporciona.

En la tercera sección, que constituye el núcleo central de la memoria, se describe el desarrollo de la aplicación, detallando todas las fases del planteamiento y la lógica general seguida al abordar cada funcionalidad.

En la cuarta sección se describe el diseño de la aplicación, explicando cuales son los elementos del interfaz gráfico (Graphic User Interface, GUI) y su funcionalidad.

En la quinta sección se describe la implementación de la aplicación, explicando en profundidad la tecnología utilizada para el desarrollo de este proyecto y la especificación técnica del mismo, bajo la perspectiva de programación Matlab (The Matworks, 2017)..

Por último se incluye un apéndice para detallar aspectos de interés y útiles tales como un manual de usuario y todos aquellos datos que proporcionan una visión general del proyecto.

1.4. Introduction

1.4.1. Preliminary Considerations

In the field of dentistry it's both important and required to rely on a tool which allows the user to achieve several estimations from an image, particularly, a mandibular radiography.

With this goal in mind, and also using our knowledge in the field of digital image processing, we have implemented an app which accomplish such objectives.

The main aim of the project is to work out the Mandibular Cortical Index (ICM), this index is a significant measure because it allows to detect early osteoporosis symptoms at the level of the maxilla and mandible.

Another objective is the study and analysis of the bone texture using panoramic radiographs, This allows the user to detect several anomalies such as dental abscess or osteoporosis. This is a direct consequence of the study and analysis of the different textures shown in the image.

In this project we have put the techniques learnt in Computational Perception to use, specially digital image processing. Although the main goal of the project is to automate the estimation of the ICM, the user is given some freedom in several steps of the execution of the app, he may be required to choose some parameters needed to make the app work, this happens because some of the steps to achieve the ICM are hard to automate.

In order to unify all the concepts and functionalities of the project we use the multi-paradigm numerical computing environment Matlab (2017), which uses a Graphical User Interface making it easier and intuitive to the user.

1.4.2. Motivation, objectives and working plan

The main objective of the project is the total automatization of the processing of radiographic images so a real problem, like the one explained below, can be simulated.

The ideal thing is that the dental office itself can easily and quickly obtain, in a matter of seconds, the necessary information from an x-ray of a patient, so that the dentist obtains the maximum possible quantitative information when it's time to issue a possible diagnosis, which is sent to the patient or informs other professionals in case of having to make a severe decision or to perform some additional specific test to the patient.

Technically, the general vision of the product consists of the development of a computer application that allows the visualization of certain data for a better diagnosis.

From the medical point of view, there are two objectives in this project:

1. Measure the Taguchi index (2010) which provides a value for determining the degree of osteoporosis of a patient, by measures of distance (in pixels) at the base of the mandible.
2. Identify different types of textures in order to discern between bony structures of other tissues and within them different levels of intensity, which could be indicative of infections or other disorders at the oral level.

From the point of view of computing and taking into account the previous considerations, the following objectives are proposed, which are subsequently transformed into requirements:

1. The user interface should be simple and intuitive.
2. Emphasize the visualization of the images x-rays in order not to lose quality, since it is the main tool of work of the client.
3. Integrate all elements in the application and make them easy to identify and use.
4. To obtain the greatest possible reliability in the measurements made as aids to the diagnosis

5. Develop a modular application that allows the incorporation of new functionalities without the need to redesign the application.

To meet the stated objectives, the following work plan is proposed,
Disaggregated by objectives and duration of the same taking into account the total duration.
Approximately eight months of the project, which makes a total of 32 weeks

Objective	Duration(weeks)
Graphic User Interface development	2
Image preprocessing and complementary features	2
Auto detection of the inferior border	10
Secondary lines layout	7
Texture description	4
Writing the memory	7 (through the duration of the project)

1.5. Contribuciones personales

Jorge Sánchez-Barriga Pons

Este proyecto empezó en 2016 a propuesta del profesor Gonzalo Pajares, del presente trabajo, y profesor de la asignatura Percepción Computacional. En clase de esta materia comentó la posibilidad de desarrollar este proyecto y mi compañero y yo decidimos presentarnos para la realización del mismo ya que tanto la asignatura como el proyecto nos despertaba mucho interés.

Una vez cerrados los casos de uso conjuntamente con el codirector del presente trabajo, a su vez profesor de Odontología comenzamos con el diseño de la aplicación, esta labor la desarrollé yo. Necesitábamos una aplicación que fuera intuitiva y sencilla, razón por la cual empecé diseñando la aplicación con un menú general donde indicaba, en primer lugar, las funcionalidades a aplicar: Cálculo del Índice Taguchi (2011) o Visualización de Texturas. En segundo lugar, según elección previa, se cargaba un panel muy parecido al actual que indicaba los pasos a seguir para aplicar la funcionalidad elegida. Visto que la división del panel entrañaba cierta complejidad, me percaté que no era necesaria la implementación de dos paneles y que se podía integrar todo en uno, tal y como se presenta actualmente. Un panel a la derecha con las funcionalidades necesarias que se activan y desactivan según si se permite el acceso a ellas. Realizado el croquis de la interfaz gráfica empecé a desarrollarla para eso utilicé el propio editor de Matlab, que de forma sencilla te permite crear una interfaz con todas sus funcionalidades y genera las estructuras de las funciones para desarrollar cada funcionalidad dentro de ellas, generando el conocido archivo de código con extensión m, propio de la herramienta Matlab.

En el apartado del código empecé desarrollando la funcionalidad de cargar imagen para mostrarla en la interfaz gráfica, donde agregué el primer pre-procesamiento de la imagen, relativa al realzado de la imagen mediante el aumento del contraste de la intensidad de los grises. La determinación de los parámetros requeridos por esta funcionalidad relativos a los valores de intensidad necesitaron de diversas pruebas, ya que era necesario encontrar los valores correctos para generar imágenes con suficiente contraste, sin llevar la imagen hacia niveles demasiado claros o oscuros.

Posteriormente realicé también el diseño e implementación de la primera parte correspondiente al cálculo del Índice de Taguchi, más concretamente todo lo que tenía que ver con el ajuste automático la primera línea que se ajusta sobre la parte inferior del borde de la mandíbula. Esta parte, y según el documento que nos habían entregado los profesores de la facultad de odontología, comenzaba con una función previamente implementada por mi compañero, concretamente la de recortar. Después desarrollé la binarización de la imagen que consiste en reemplazar todos los valores superiores a un umbral por el valor 1 (blanco) y los demás por 0 (negro), esto nos permite distinguir perfectamente las diferentes regiones de la imagen según dicho umbral. Una vez diferenciadas las regiones de la imagen y limpiadas las manchas innecesarias implementé la función para detectar los bordes de la imagen, usamos la aproximación correspondiente a la primera derivada por el método de Sobel, esta función devuelve todos los bordes que sean mayores que un rango introducido por parámetro, tras diversas pruebas de ensayo error, obtuve el rango de valores correcto, pues no siempre se detectaba bien los bordes.

Posteriormente implementé el dibujo de la línea ajustada en sí, para ello debemos segmentar la imagen en regiones con el fin de separar los bordes que habíamos detectado anteriormente, esto lo conseguí mediante técnicas basadas en transformación del color, primero transformando la imagen a valores sin signo de ocho bits para separar los canales de la imagen y después determinando las regiones en la imagen utilizando el sistema de color CIE-XYZ. Posteriormente, y ya separadas las diferentes regiones de la imagen, desarrollé el procedimiento para detectar todas las líneas potenciales de la imagen usando la transformada de Hough que permite encontrar todo tipo de ecuaciones paramétricas que puedan ser expresadas matemáticamente, en nuestras líneas rectas. Esta función por sí misma nos devuelve muchas rectas de las que solo nos interesa una, por ello desarrollé y adapté el procedimiento para unir líneas similares estableciendo la longitud mínima de las mismas, fusionando líneas con similares (ρ, θ) . Con esto ya detectamos correctamente la recta inferior de la mandíbula.

Hay que resaltar que el código del proyecto se ha desarrollado en tres fases, en un principio ninguna línea se iba a detectar automáticamente, todo lo hacía manualmente el usuario, esto se planteó así para tomar contacto con la aplicación, en esta fase el objetivo principal consistía tanto en el diseño como la manera de pintar las líneas junto con la búsqueda de información necesaria. La siguiente fase consistía en automatizar la detección de la recta tangencial al borde inferior de la mandíbula, tuvimos muchos problemas pues no se había implementado un tratamiento previo sobre la imagen para detectar los bordes, concretamente la detección del borde inferior mediante la función de Sobel (2015).

Respecto a la memoria realicé el apartado relativo a la descripción de la implementación en un principio fui explicando cada función por separado con todos sus valores y argumentos necesarios, este trabajo consistió en buscar en la documentación de Matlab(2017) así como en la bibliografía especializada. De acuerdo con las decisiones tomadas en las distintas reuniones de seguimiento y de acuerdo con el avance de los desarrollos, se planteó incluir el código explicado en un Anexo para lo que tuve que rehacer el apartado correspondiente y separar todas las líneas de código con referencia a las mismas en la sección de implementación.

Por otra parte también desarrollé la sección de introducción, evolucionando en el tiempo hasta la configuración actual. También realicé el apartado de objetivos y plan de trabajo. Elaboré el anexo de Manual de usuario explicando paso por paso cómo realizar las funcionalidades de la aplicación.

Además revisé el código dándole formato y comentando cada función para hacer más fácil su comprensión.

Marcos Rebollo Fernández

Nuestro tutor y profesor Gonzalo Pajares nos propuso realizar varios proyectos relacionados con el tratamiento de imágenes y los conceptos vistos en la asignatura de Percepción Computacional, tanto a mi compañero como a mí nos llamó la atención el poder orientar el trabajo de fin de grado hacia la realización de una aplicación aplicada al campo de la odontología y que además pudieran tener una aplicación real.

Una vez delimitados el alcance y las principales funcionalidades con las que debía contar la aplicación, nos pusimos a plantear el diseño y desarrollo de la misma.

Tras una primera toma de contacto con Matlab, revisando la documentación bibliográfica disponible junto la proporcionada por el profesor tutor, realicé la parte del diseño e implementación de carga de radiografías panorámicas y su visualización en el panel de la interfaz correspondiente a la aplicación.

Una vez mostrada la imagen seleccionada en la interfaz gráfica, desarrollé las funcionalidades básicas para poder manipularla, particularmente el recorte de imágenes, que permite al usuario ampliar regiones seleccionadas.

También la opción de restaurar la imagen, que permite deshacer todos los cambios y regresar a la imagen original, y por último el panel de instrucciones que indica al usuario los pasos a seguir según la funcionalidad elegida. Se empezó desarrollando la primera funcionalidad de la aplicación, el cálculo del índice cortical mandibular, también conocido como índice taguchi. En una primera versión la aplicación era totalmente manual, haciéndolo poco preciso y molesto para el usuario, por lo que se decidió automatizar el proceso mediante el reconocimiento de bordes.

En esta segunda versión me encargué de colaborar en la obtención automática de la primera recta, que se ajusta a la parte inferior de la mandíbula, una vez transformada la imagen original a una imagen binaria de dos canales (blanco y negro), realicé el diseño e implementación de las funcionalidades orientadas a la limpieza de la misma, eliminando las componentes conexas de píxeles que fueran innecesarios, sin aportar ningún valor añadido al objetivo final, esto es la detección del borde inferior de la mandíbula y trazar la primera recta.

Debido a que la calidad de las radiografías hacía inviable el poder reconocer el borde superior mandibular opté por la opción de dejar en manos del usuario el trazar la segunda recta.

En un principio el usuario tenía absoluta libertad para trazar la recta como quisiera, esta idea se descartó rápidamente ya que una de las especificaciones exigía que esta segunda recta fuera paralela a la primera. Descartada la opción de trazado libre, el usuario únicamente tiene que elegir un punto por el que trazar esta recta, contando con dos puntos de la primera recta implementé el cálculo de la pendiente de la misma, de forma que utilizando el punto seleccionado por el usuario y las ecuaciones de la recta implementé la funcionalidad para el trazado de esta segunda recta.

La última recta debe ser perpendicular a las anteriores y trazarse de forma que pase por el punto conocido como agujero mentoniano de la mandíbula.

De manera similar a lo ocurrido con la segunda recta paralela, la identificación del agujero mentoniano sobre la imagen era difícil de detectar en las radiografías por lo que finalmente se asignó al usuario la tarea de seleccionar la ubicación del mismo.

Realicé el diseño e implementación de las funcionalidades relativas al trazado de estas rectas.

Una vez calculadas y trazadas todas las tres rectas mencionadas, calculé los puntos de intersección entre las mismas, esta distancia entre puntos determina el valor del índice cortical mandibular.

Tras terminar la primera funcionalidad relativa al cálculo del mencionado índice, me dediqué a completar la segunda, esto es la descripción de texturas.

La descripción de texturas constituye otro método como ayuda al diagnóstico para determinar determinadas estructuras de los tejidos en el paciente, en primer lugar diseñé la aplicación de forma que el usuario pudiera seleccionar una región de interés sobre la que realizar la descripción de texturas.

Después se realiza el cálculo de la desviación típica de cada píxel en función de una región de vecindad alrededor del mismo, diseñé la forma de almacenar los datos de las desviaciones típicas en una matriz que posteriormente se mostraba como una imagen con dichos datos escalados en un determinado rango de colores, permitiendo la clara distinción a simple vista de los contrastes de intensidad de la textura ósea frente a los tejidos blandos.

En cuanto al desarrollo de la memoria del proyecto, me encargué de realizar las traducciones al inglés de los apartados pertinentes, además de las secciones del primer capítulo que tratan sobre la estructura general de la memoria. Por otra parte, me encargué de la estructuración de la memoria tal y como se ha presentado finalmente, así como de la descripción en líneas generales de las funcionalidades contenidas en la aplicación final. También me encargué íntegramente de la redacción del capítulo de desarrollo, en el cual explico detalladamente los diferentes procedimientos seguidos para conseguir las diferentes funcionalidades deseadas en la aplicación, además de una explicación sobre sus fundamentos teóricos.

En el apartado de diseño realicé los diferentes diagramas de clase, casos de uso y diagramas de actividad finalmente incorporados en la memoria, partiendo de un escenario habitual en el uso de la aplicación.

Por último me encargué de realizar las diferentes capturas de pantalla de la aplicación, que aparecen en la memoria junto con su descripción. Finalmente fui el encargado de coordinar la estructuración y redacción de la memoria.

2. Funcionalidades de la aplicación.

Como se ha indicado previamente, la aplicación tiene como objetivo la automatización de dos procedimientos de análisis de imágenes de radiografías como ayuda al diagnóstico en odontología. Esto supone la introducción de tecnologías informáticas en el campo de la medicina para facilitar las tareas de diagnosis en un mundo en continua evolución.

El objetivo principal consiste en poder monitorizar una serie de datos, radiografías mandibulares, para que, mediante una serie de procedimientos, aplicando técnicas de procesamiento de imágenes e Inteligencia Artificial, puedan suministrar al usuario datos de especial relevancia como ayuda a la toma de decisiones y a un diagnóstico más rápido y eficaz que pueda prevenir y ayudar en el tratamiento de algunas enfermedades bucales e incluso, como en el caso que nos ocupa, ampliando el diagnóstico de cara a la detección de la osteoporosis como una patología que puede detectarse por el análisis de la mandíbula, mediante la determinación del conocido como índice de Taguchi (2011).

Las imágenes utilizadas para el desarrollo de la aplicación fueron proporcionadas por la facultad de odontología de la UCM. Con ellas se pudo empezar a trabajar en el proyecto, mediante el correspondiente diseño de la interfaz gráfica.

Posteriormente, se decidió la disposición de los elementos para que fueran lo más fáciles de interpretar por parte del usuario, en este caso, especialistas en odontología. A partir de este momento se comenzó a trabajar en la funcionalidad de la aplicación. Como se ha explicado previamente, son dos las funcionalidades principales desarrolladas con el fin de dar respuesta al cálculo del mencionado índice de Taguchi (2011) y la descripción de las texturas subyacentes en la imagen bajo análisis. Ambas se describen a continuación.

2.1. Cálculo del índice de Taguchi.

2.1.1. Generalidades sobre el índice de Taguchi.

La osteoporosis es una enfermedad que viene dada por una pérdida en la densidad ósea y un deterioro de lo que se conoce como microarquitectura del tejido óseo, el cual puede conllevar fracturas. El riesgo de sufrir osteoporosis es más elevado en mujeres (40%) que en hombres (13%). Entre los principales determinantes de la osteoporosis se incluye una deficiencia de estrógeno en mujeres, bajo índice de masa corporal, consumo de tabaco y alcohol, escaso consumo de calcio e inactividad física, entre otras.

Para detectar signos de osteoporosis a nivel del maxilar y la mandíbula se emplea habitualmente el ICM, o índice Taguchi (2011), que consiste en una clasificación ordinal, tomando en consideración las características de la base de la mandíbula, en el área distal al foramen mentoniano. Este índice se calcula mediante la observación de la zona mandibular en una radiografía panorámica, trazando una recta tangencial al borde inferior de la mandíbula, otra paralela a ésta por el borde superior de la cortical mandibular y por último una recta perpendicular a estas dos que desciende desde el borde inferior del agujero mentoniano e intersecta a las dos rectas paralelas en dos puntos de corte, la distancia entre dichos puntos de corte determina el valor del ICM (Abhyankar y Shriram, 2013; Hastar y col., 2011). Tal y como se aprecia en la figura 1; la recta 1 es tangencial al borde inferior de la mandíbula, su paralela lo es por el borde superior, y la recta 2 es perpendicular a ambas y atraviesa el agujero mentoniano, la distancia C entre los puntos de corte es el valor del ICM.

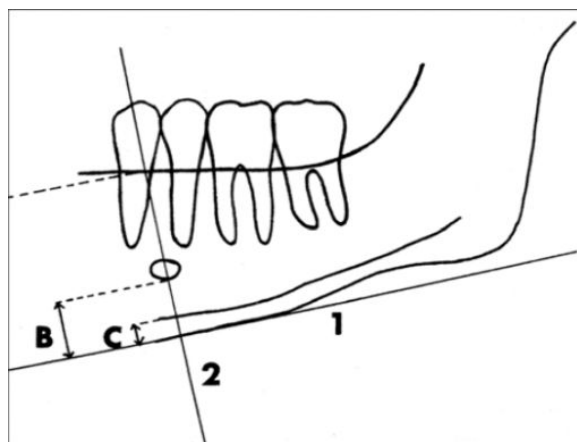


Figura 1. Diagrama del cálculo del ICM (imagen tomada de Marandi y col. , 2010)

2.1.2. Consideraciones sobre el cálculo del Índice de Taguchi.

Una vez definido el ICM, el planteamiento consiste en determinar la distancia perpendicular entre los bordes interiores y exteriores de la mandíbula. Con tal propósito se realizan una serie de procesamiento previos sobre las imágenes según los conceptos establecidos en (Abhyankar y Shriram (2013) y Marandi y col. (2010).

En primer lugar se realiza un preprocesamiento sobre la imagen, realizando las zonas de interés a estudiar para poder determinar e identificar con la máxima precisión posible los bordes o perfiles que determinan los bordes mandibulares, ya que con la radiografía original su definición no resulta suficientemente evidente. Este paso consiste en transformar la imagen a escala de grises, binarizar y limpiar el ruido.

Una vez aplicado el preprocesamiento, se detectan los bordes mediante una aproximación basada en el concepto primera derivada mediante la implementación del conocido método de Sobel (2015), como se detalla posteriormente. Una vez se detecta el borde inferior de la mandíbula, se procede a ajustar una línea recta por el método de mínimos cuadrados. Estos dos procesos de tratamiento de imágenes se encuentran descritos en (Pajares y Cruz, 2007). Una vez ajustada al borde de la mandíbula la línea inferior, se procede al trazado de una paralela, tratando de ajustarla al borde mandibular superior.

Una vez trazadas las dos líneas paralelas se traza una perpendicular a ambas pasando por el agujero mentoniano de la mandíbula. La distancia entre los dos puntos de corte de esta perpendicular sobre las dos líneas paralelas es lo que se conoce como ICM.

Para facilitar al profesional de odontología el uso de la aplicación se permite hacer un recorte sobre la radiografía, para resaltar con mayor precisión los detalles de la imagen dentro del recorte y así poder seleccionar el área sobre la que se calcula de forma automática el ICM. Realmente, el método desarrollado es semiautomático, ya que el usuario solamente tiene que seleccionar el área de estudio y un punto del borde superior de la mandíbula para que se realice el cálculo, ahora ya sí de forma automática.

2.2. Descripción de Texturas.

2.2.1. Generalidades sobre el cálculo de texturas.

En las radiografías panorámicas mandibulares, el análisis de textura constituye una herramienta de gran utilidad para evaluar a los pacientes con posible osteoporosis, considerándose uno de los métodos más útiles y precisos para el diagnóstico de dicha enfermedad.

Existen en la literatura diferentes técnicas de medición de texturas y análisis fractal para diferenciar individuos que sufren osteoporosis de los que no la sufren (Hastar y col., 2011).

Tales parámetros de textura ósea derivadas de radiografías han puesto de manifiesto una fuerte relación con la Densidad Media Ósea (DMO).

En los estudios clínicos, la radiografía digital de alta resolución es una técnica de imagen ampliamente disponible con la precisión necesaria para evaluar la microarquitectura ósea.

Se ha demostrado que la combinación de los parámetros de textura y la densidad mineral ósea supone una mejora para la evaluación del riesgo de fractura en comparación con el uso de parámetros de textura o la DMO por sí solos.

La combinación de parámetros de la estructura morfológicas y la DMO, han demostrado correlaciones más altas con la fuerza del hueso.

2.2.2. Consideraciones sobre el cálculo de texturas.

La aplicación incluye una funcionalidad para el análisis de la textura ósea con el objetivo de determinar si las mediciones individuales (ICM) o una combinación de características texturales y de la anchura cortical mandibular (derivadas de radiografías panorámicas dentales digitales) son más útiles en la evaluación de la osteoporosis.

Mediante dicha funcionalidad, el usuario selecciona un área rectangular sobre la que quiera realizar el cálculo de las texturas, tras lo cual se muestra de forma automática, a través de la interfaz de la aplicación dicha región de interés con los valores de intensidad de la textura claramente diferenciados.

Existen diversos descriptores de textura (Pajares y Cruz, 2007), de los que se ha desarrollado uno de naturaleza estadística basado en el cómputo de la varianza de los niveles de intensidad de la imagen sobre ventanas de un determinado tamaño (ancho y alto). Se trata de un descriptor de naturaleza dinámico local en tanto en cuanto la varianza se calcula sobre las mencionadas ventanas, sin tener en cuenta la imagen en su conjunto.

Además de lo indicado previamente, el análisis de textura resulta de una ayuda para un posterior estudio sobre la imagen que permite determinar posibles problemas sobre la estructura ósea del paciente.

3. Desarrollo

Según las consideraciones anteriores, y teniendo en cuenta que se trata de una aplicación que se diseña por primera vez, el desarrollo del proyecto contempla dos fases principales, que coinciden exactamente con el cálculo de ICM y descripción de texturas.

A continuación se describen los detalles, de cada una de las dos metodologías mencionadas.

3.1. Pre-procesamiento.

Se permite la carga de radiografías panorámicas en cualquier formato de imagen, incluido el conocido como portable network graphics (con extensión .png) en la aplicación mediante una interfaz gráfica amigable al usuario, que se describe con detalle en la sección cuatro, dedicada al diseño de la aplicación.

Tras la carga de la radiografía deseada ésta se transforma a una imagen en escala de grises mediante la transformación del modelo de color RGB, que es el original de las imágenes utilizadas, al modelo HSV (Pajares y Cruz, 2007). De esta forma, se consigue un aumento del contraste de la imagen lo que permite posteriormente realizar cálculos sobre ella con mayor facilidad.

3.2. Cálculo del índice de Taguchi.

Como se ha explicado anteriormente el cálculo del ICM se basa en medir la distancia entre dos rectas paralelas, la primera recta se genera automáticamente calculando la detección de bordes sobre la imagen y trazando la correspondiente recta asociada tangencialmente a la zona del cortical mandibular inferior.

La segunda recta, que debe ser paralela a ésta, se traza siguiendo el borde superior de la mandíbula, mediante la guía establecida por un punto. En definitiva, en este último caso se trata de trazar la paralela a una recta dada que pasa por un punto. Concretamente, este punto pertenece a la línea que define el borde superior de la mandíbula

La distancia entre los dos puntos de corte, que genera la recta perpendicular a las dos rectas paralelas, y que pasa por el agujero mentoniano, permite determinar el ICM. La aplicación muestra este valor a través de la interfaz de usuario, de suerte que el profesional en odontología posee esta información de forma instantánea.

Los cálculos se obtienen mediante la aplicación de la transformada de Hough (Gonzalez y col., 2004; Pajares y Cruz, 2007), que permite el ajuste de una línea recta dada la alineación de puntos definidos por los puntos de borde que se extraen tras la aplicación de los operadores de Sobel (Sobel 2015; Gonzalez y col., 2004; Pajares y Cruz, 2007)

3.2.1. Detección automática del borde

Para realizar la detección automática del borde, se realiza un conjunto de operaciones secuenciadas en el tiempo, que se exponen a continuación.

3.2.1.1. Recorte de la imagen.

Una de las opciones del menú mostrado en la interfaz de la aplicación permite realizar un recorte rectangular de la imagen original para ampliar una región de interés y mostrarla con más detalle al usuario para su posterior análisis y estudio.

3.2.1.2. Binarización de la imagen.

La aplicación obtiene una imagen binaria a partir del recorte en escala de grises, es decir una imagen con únicamente dos valores posibles para cada píxel (blanco ó negro en este caso).

En el proceso de binarizar la imagen correctamente se necesita calcular un umbral óptimo que defina la característica blanca o negra del píxel, para ello se recurre al método de Otsu (1979).

La mayoría de las técnicas de umbralización se basan en estadísticas sobre el histograma unidimensional de intensidades, en particular el método de Otsu elige el umbral óptimo maximizando la varianza entre clases mediante una búsqueda exhaustiva sobre el histograma de la imagen..

Una imagen es una función bidimensional de la intensidad del nivel de gris o intensidad, y contiene **N** píxeles cuyos niveles de gris se encuentran entre 1 y L. El número de píxeles con nivel de gris *i* se denota como *f_i*, y la probabilidad de ocurrencia del nivel de gris *i* en la imagen está dada por

$$P_i = \frac{f_i}{N} \quad (1)$$

En este caso, en la umbralización en dos niveles de una imagen (también conocida como binarización), los píxeles son divididos en dos clases: C1, con niveles de gris [1, ..., *t*]; y C2, con niveles de gris [*t*+1, ..., L]. Entonces, la distribución de probabilidad de los niveles de gris para las dos clases son:

$$C_1 : \frac{P_1}{\omega_1(t)}, \dots, \frac{P_t}{\omega_1(t)} \quad (2)$$

$$C_2 : \frac{P_{t+1}}{\omega_2(t)}, \frac{P_{t+2}}{\omega_2(t)}, \dots, \frac{P_L}{\omega_2(t)} \quad (3)$$

donde

$$\omega_1(t) = \sum_{i=1}^t p_i \quad \omega_2(t) = \sum_{i=t+1}^L p_i \quad (4)$$

También, la media para la clase C1 y la clase C2 es:

$$\mu_1 = \sum_{i=1}^t \frac{i \cdot P_i}{\omega_1(t)} \quad \mu_2 = \sum_{i=t+1}^L \frac{i \cdot P_i}{\omega_2(t)} \quad (5)$$

Sea μ_T la intensidad media de toda la imagen. Es fácil demostrar que:

$$\omega_1 \cdot \mu_1 + \omega_2 \cdot \mu_2 = \mu_T \quad \omega_1 + \omega_2 = 1 \quad (6)$$

3.2.1.3. Limpieza de ruido

Una vez obtenida la imagen binaria es conveniente realizar una operación de filtrado sobre la misma y eliminar píxeles que sean innecesarios.

Para ello se genera otra imagen binaria a partir de la original eliminando los píxeles aislados, es decir se eliminan las componentes conexas de píxeles blancos que estén formadas por un número inferior a p píxeles (siendo p un límite preestablecido).

Al eliminar estas componentes consideradas como ruido, y que no forman parte del borde mandibular inferior que se pretende aislar, se obtiene una nueva imagen binaria más limpia que la original.

3.2.1.4. Detección de bordes.

Los bordes de una imagen digital se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos.

Para detectar los bordes de la imagen binarizada se utiliza habitualmente el operador de Sobel (2015). El operador Sobel es una aproximación a la primera derivada de los niveles de grises de la imagen, calcula una aproximación al gradiente de la función de intensidad de cada punto en la imagen, determinando en cada punto la magnitud del mayor cambio posible.

Este operador representa una aproximación rápida aunque algo imprecisa del gradiente de la imagen; utilizando valores de intensidad en una región 3x3 (conocida como matriz de vecindad) alrededor de cada píxel, se calcula el gradiente correspondiente.

x-componente		
-1	0	1
-2	0	2
-1	0	1

y-componente		
1	2	1
0	0	0
-1	-2	-1

En cada punto de la imagen, los resultados de las aproximaciones de los gradientes horizontal (x-componente) y vertical (y-componente) pueden ser combinados para obtener la magnitud del gradiente (G) y su dirección (θ).

$$G = \sqrt{G_x^2 + G_y^2} \quad (7)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (8)$$

3.2.2. Generación de la primera recta.

3.2.2.1. Transformada de Hough.

Para generar la primera recta tangencial al borde inferior de la mandíbula se usa la transformada de Hough para detección de rectas.

Se trata de un método propuesto por Hough (1962) para encontrar la ecuación de una línea que pase por un conjunto de n puntos en el plano xy .

Tal y como se explica en MartinSanz P. G. (2008), una línea que une una secuencia de puntos o píxeles puede expresarse en forma paramétrica como aparece en la siguiente expresión, el objetivo es por tanto evaluar los parámetros a y b , que definen la ecuación de la recta.

$$y = ax + b \quad (9)$$

Considérese un punto (x_i, y_i) expresado en coordenadas de la imagen,, existe un número infinito de líneas que pasan por dicho punto y que satisfacen todas ellas la ecuación $y_i = ax_i + b$ para valores variables de a y b . Si escribimos la ecuación como $b = -x_i a + y_i$ y consideramos el plano ab también llamado *espacio de parámetros*, tendríamos la ecuación de una única línea para un valor fijo de (x_i, y_i) . Un segundo punto (x_j, y_j) tendrá también una línea en el espacio de parámetros asociada y esta línea intercepta con la línea asociada a (x_i, y_i) en (a', b') , donde a' es la pendiente y b' la altura a la que la recta que contiene a los dos puntos (x_i, y_i) y (x_j, y_j) corta al eje xy . De hecho todos los puntos contenidos en esta línea tendrán rectas asociadas en el espacio de parámetros que se cortarán en (a', b') .

Esto hace posible realizar agrupaciones de los puntos que pertenecen a los bordes de posibles figuras a través de un procedimiento de votación sobre un conjunto de figuras parametrizadas.(Martínez (2017)).

Uno de los inconvenientes para la obtención de los parámetros a y b estriba en la necesidad de delimitar el rango de sus valores, algo que no siempre resulta trivial. Para solventar este problema se recurre a la representación paramétrica en coordenadas polares, de suerte que el ángulo θ varía en el rango $[-180^\circ, 180^\circ]$ y ρ entre cero y el valor la diagonal de la imagen.

$$\rho = x \cos\theta + y \sin\theta \quad (10)$$

Para un punto arbitrario en la imagen con coordenadas (x,y) , las rectas que pasan por ese punto son los pares (ρ,θ) con $\rho = x \cos\theta + y \sin\theta$ donde ρ es la distancia entre la línea y el origen y θ el ángulo que forma la recta con el eje x.. Esto corresponde a una curva sinusoidal en el espacio (ρ,θ) que es única para ese punto. Si las curvas correspondientes a dos puntos se interceptan, el punto de intersección en el espacio de Hough corresponde a una línea en el espacio de la imagen que pasa por estos dos puntos.

El algoritmo de la transformada de Hough usa una matriz, llamada acumulador de celdas, cuya dimensión en el caso de una recta será dos, correspondientes a los valores cuantificados para (ρ,θ) . Para construir el acumulador es necesario discretizar los parámetros que lo determinan, esto es ρ y θ en los rangos indicados previamente. Cada celda del acumulador está indexada por valores de los parámetros ρ y θ en los rangos indicados previamente.

Cada punto en la imagen incrementa el valor de las celdas del acumulador de suerte que los puntos que pertenecen a una misma recta incrementan el valor de una única celda. De esta forma dicha celda posee un valor alto en el acumulador. Las rectas se identifican precisamente por la existencia de celdas con valores de acumulación altos. Si bien, es necesario determinar lo que se considera alto, fijando un valor de umbral a tal efecto para eliminar posibles valores de acumulación que no representan realmente una recta. En la aplicación desarrollada este valor se ha fijado a 200, que se corresponde con un valor apropiado en función del recorte de la imagen que se realiza en la primera parte del proceso.

Mediante la secuencia de procesos indicada previamente se obtiene de forma automática la primera recta tangencial al borde inferior de la mandíbula.

3.2.3. Generación de la recta paralela.

Una vez obtenida la recta tangencial al borde inferior de la mandíbula se genera la recta tangencial al borde superior de la mandíbula y paralela a la anterior.

Dado que la calidad de la radiografía no es suficientemente alta como para permitir la identificación del borde de forma automática, el usuario de la aplicación debe indicar con el cursor un punto de la radiografía que pertenezca al borde superior mandibular y sobre el que trazar la recta paralela.

De forma que la recta paralela se reduce al sencillo problema matemático de obtener la ecuación de una recta paralela a otra dada y que pase por un punto dado. El procedimiento para calcular esta segunda recta paralela se describe a continuación.

3.2.3.1. Cálculo de la pendiente.

Se calcula la pendiente de la primera recta tangencial al borde inferior de la mandíbula obtenida de forma automática, que será coincidente con la pendiente de la segunda recta al tratarse de dos rectas paralelas.

Teniendo dos puntos cualesquiera pertenecientes a la primera recta obtenida de forma automática, en este caso el punto (x_1, y_1) y el punto (x_2, y_2) , se calcula la pendiente (m):

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \quad (11)$$

3.2.3.2. Cálculo de coordenadas.

Conocida la pendiente de la nueva recta y las coordenadas del punto, indicado por el usuario, se calcula mediante la ecuación lineal de la recta el término independiente **b**, que resulta ser el punto de corte de la recta con el eje de ordenadas.

$$y = mx + b \quad (12)$$

A continuación utilizando la ecuación paramétrica de la recta se calcula un nuevo punto perteneciente a la nueva recta y se traza ésta, situándose tangencial al borde superior de la mandíbula y paralela a la recta tangencial al borde inferior.

3.2.4. Generación de la recta perpendicular.

Una vez calculadas las rectas paralelas tangenciales al borde inferior y superior de la mandíbula, solamente falta trazar una línea perpendicular a ambas rectas paralelas pasando por el punto correspondiente al agujero mentoniano de la mandíbula para finalmente calcular la distancia entre los puntos de corte.

3.2.4.1. Cálculo de la pendiente.

Se trata de un cálculo matemático sencillo, ya que la pendiente de la recta perpendicular a una dada con pendiente m recta es la inversa de la pendiente cambiada de signo de la recta dada.

$$m' = \frac{-1}{m} \quad (13)$$

3.2.4.2. Cálculo de coordenadas.

Antes de continuar con la descripción del proceso, conviene realizar las siguientes consideraciones en relación al agujero mentoniano en la imagen, que determina el trazado de la recta perpendicular a las dos rectas paralelas asociadas a los bordes de la mandíbula. El agujero mentoniano, resulta fácil de distinguir en el hueso físico correspondiente a la mandíbula, si bien en las radiografías su identificación resulta de extrema dificultad, por lo que no se ha podido determinar el conjunto de técnicas de procesamiento de imágenes requeridas para su identificación de forma automática, figura 2.

Por este motivo, es el propio usuario el que debe indicar, basado en su experiencia, dónde se sitúa mediante el cursor.

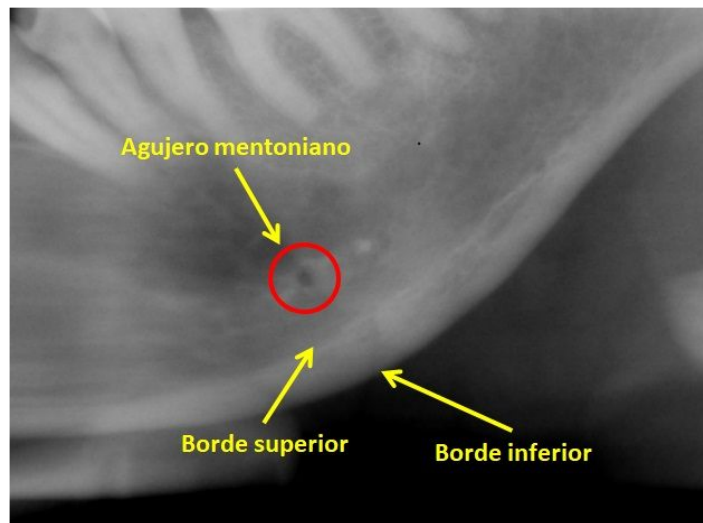


Figura 2. Identificación del agujero mentoniano.

Por consiguiente, una vez identificado y seleccionado dicho agujero por el usuario a través de la interfaz de la aplicación se dispone de las coordenadas de imagen de dicho punto, momento a partir del cual se utiliza las ecuaciones lineales de la recta para trazar la recta perpendicular que pasa por este agujero, como se ha explicado previamente.

3.2.5. Intersecciones entre rectas.

Una vez calculadas las rectas necesarias sobre la radiografía, es necesario calcular la distancia entre los puntos de corte de la perpendicular con las rectas paralelas, lo que determinará finalmente el valor del ICM.

3.2.5.1. Punto de intersección entre rectas.

Conociendo las ecuaciones lineales de las rectas paralelas, según la ecuación (14).

$$y = ax + c \qquad y = bx + d \qquad (14)$$

se sabe que en el punto de corte entre ambas su coordenada y coincide, por lo que igualando dicha coordenada se tiene:

$$ax - bx = d - c \qquad (15)$$

Despejando se obtiene el valor de la coordenada x ; de forma que sustituyendo el valor de x en cualquiera de las ecuaciones lineales se obtiene el valor de la coordenada y . Por lo que un punto de corte viene determinado por la expresión:

$$P\left(\frac{d-c}{a-b}, \frac{ad-bc}{a-b}\right) \qquad (16)$$

3.2.6. Distancia entre los puntos generados (Índice Taguchi).

Llegados a este punto, sólo falta medir la distancia entre los dos puntos de corte obtenidos entre la recta perpendicular a las paralelas de éstas, a saber: P_1 (punto de corte entre la perpendicular y la recta ajustada al borde inferior de la mandíbula) y P_2 (punto de corte entre la perpendicular y la paralela a la anterior, trazada por el borde superior de la mandíbula). Para ello se mide la distancia euclídea entre dos puntos, expresada en píxeles de la imagen.

La distancia euclidiana entre dos puntos P_1 y P_2 , de coordenadas cartesianas (x_1, y_1) y (x_2, y_2) respectivamente, viene dada por la siguiente expresión:

$$D(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (17)$$

3.3. Descripción de texturas.

3.3.1. Pre-procesamiento.

Como se ha indicado previamente, el análisis de textura usando radiografías de estructuras óseas y su posterior estudio está considerado uno de los métodos útiles para detectar diversas estructuras en los tejidos óseos o blandos a través del análisis de las imágenes. La distinta estructura de los mencionados tejidos, en este caso óseos, también puede ser indicativo de anomalías derivadas de la osteoporosis.

Pues bien, para este análisis la aplicación permite al usuario seleccionar, mediante un recorte rectangular sobre la imagen original, una región de interés sobre la que se aplicará el cálculo de la textura. En este caso, el descriptor de textura utilizado es la varianza o equivalentemente la desviación típica del nivel de gris (Pajares y Cruz, 2007).

3.3.2. Cálculo de la desviación típica.

Estadísticamente, la desviación típica (σ) es una medida del grado de dispersión de los

datos en una muestra con respecto al valor promedio, se obtiene mediante la raíz cuadrada de la varianza.

Desde el punto de vista del tratamiento de imágenes, se calcula la desviación típica en cada píxel seleccionando alrededor de él una región de interés de dimensión $n \times n$, en el caso que nos ocupa de tamaño 3×3 . A partir de esta región se obtienen los valores de intensidad en la misma, con los que se calcula σ . Estos valores así obtenidos se almacenan en una matriz de las mismas dimensiones que la imagen original.

La aplicación permite la visualización de estos valores en la correspondiente ventana, apareciendo los valores coloreados según la escala en el rango de valores obtenidos.

Dado que la desviación típica mide las desviaciones de los valores respecto de la media, valores bajos de ésta expresan desviaciones pequeñas y por tanto regiones con valores de intensidad más o menos homogéneos. Por ejemplo para un área de intensidad constante, pongamos por caso 108, la media sería precisamente este valor y σ sería cero. Por el contrario, valores altos de σ indican fuertes variaciones de las intensidades en la región y por tanto no homogéneas. Por consiguiente, la inspección visual de los valores de la textura así obtenidos permiten determinar la variabilidad de las regiones analizadas, sirviendo como ayuda al diagnóstico.

4. Diseño

El diseño de la interfaz de usuario, se ha realizado utilizando la aplicación Matlab (2017) con su correspondiente herramienta gráfica.

Desde el punto de vista del diseño se ha decidido que la aplicación cuenta con tres partes básicas:

- Una ventana principal que ocupe gran parte de la ventana principal para visualizar en ella las radiografías que se vayan a tratar.
- Un menú en la parte derecha con el que el usuario puede acceder a todas las funcionalidades de la aplicación. Y además con un apartado de instrucciones para que el usuario sepa qué hacer en cada momento.
- En la parte inferior cuenta con un campo para visualizar los resultados obtenidos tras el cómputo del ICM, así mismo los resultados obtenidos tras el análisis de la descripción de texturas se muestran en una ventana auxiliar aparte.

En la figura 3 se muestra el correspondiente diagrama de clases, correspondientes al diseño de la aplicación.

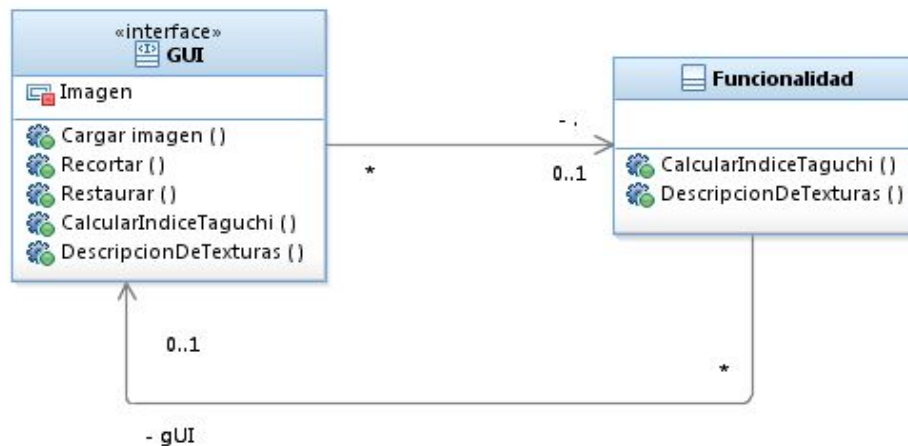


Figura 3. Diagrama de clases

A continuación se detalla un caso de uso de una funcionalidad representativa de la aplicación, particularmente se trata del análisis de la descripción de texturas.

Caso de uso de la funcionalidad descripción de texturas:

Título	Descripción de texturas.
Descripción	El usuario necesita realizar la descripción de texturas.
Actor Principal	Usuario.
Precondiciones	<ol style="list-style-type: none"> 1. La radiografía panorámica deberá estar cargada en la aplicación. 2. El usuario ha seleccionado el botón descripción de texturas. 3. No debe existir ninguna otra operación en curso.
Postcondiciones	<ol style="list-style-type: none"> 1. La aplicación mostrará por la GUI la región de interés con su descripción de texturas.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario selecciona en la GUI el botón "<i>Descripción de texturas</i>". 2. El usuario selecciona la región de interés. 3. Se realiza el cálculo de la desviación típica.
Frecuencia de uso / Prioridad	Alta / Alta
Datos entrada	Imagen radiográfica panorámica.
Datos salida	Imagen con los contrastes de intensidad en la textura ósea.

La figura 4 muestra el diagrama de actividad relativa a de la funcionalidad denominada descripción de texturas.

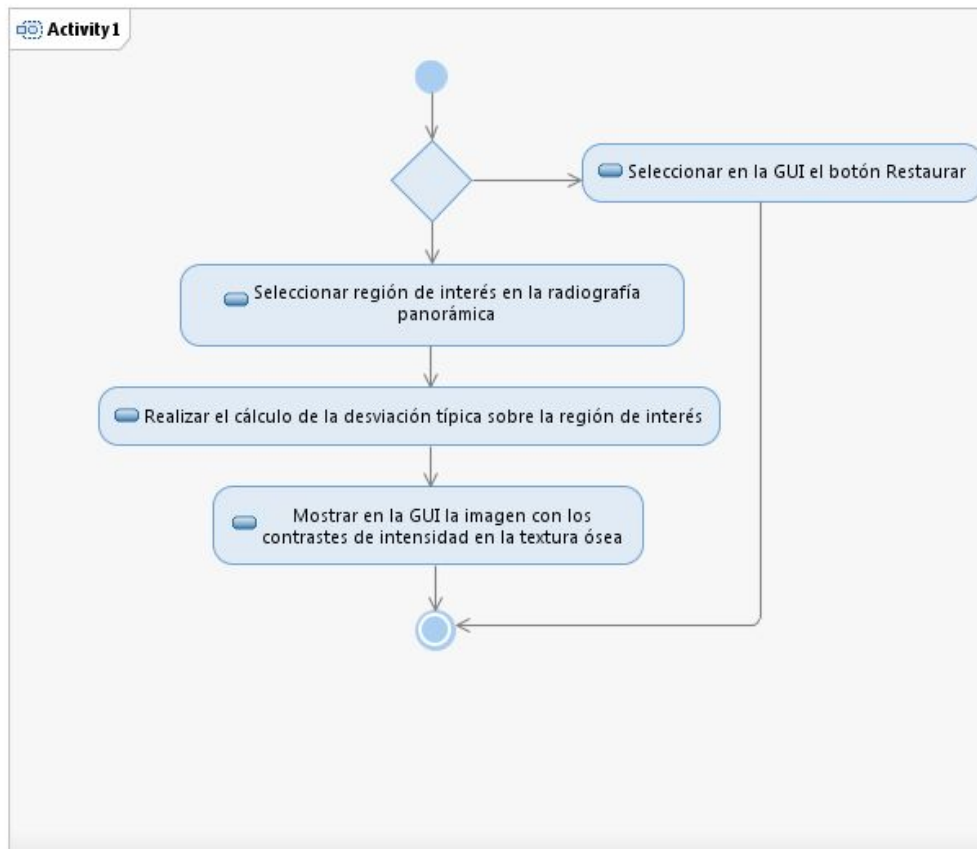


Figura 4. Diagrama de actividad

5. Implementación y resultados.

Esta sección contiene los detalles sobre la implementación de la aplicación tomando como referencia el lenguaje Matlab utilizado con tal propósito. Se describen los procedimientos relativos al cálculo del índice de Taguchi y las texturas, incluyendo los detalles del código más relevantes e ilustrando el proceso con los resultados de las imágenes procesadas en cada caso.

5.1. Pre-procesamiento.

Una vez cargada la imagen original en la aplicación, ésta se transforma a una nueva escala de grises mediante la siguiente función **imgadjust** (Código Anexo - punto 1).

Esta función mapea los valores de intensidad en la escala de grises de la imagen original a una nueva, para aumentar el contraste de grises de la imagen resultante, en función de los dos parámetros de ajuste de intensidad (**[low_in; high_in]** y **[low_out; high_out]**).

La figura 5(a) muestra la imagen original de la radiografía sin tratar, la figura 5(b) muestra el resultado de realizar una transformación de cambio de escala, observándose un aumento del contraste con respecto a la imagen original.

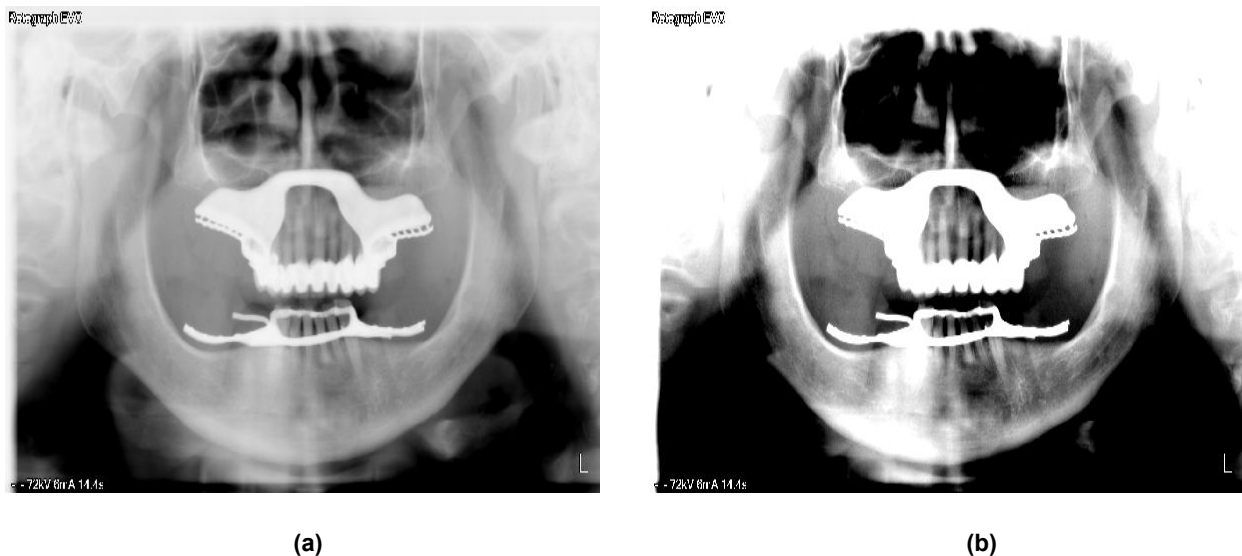


Figura 5. (a) Imagen original; (b) Imagen transformada a una nueva escala de grises

En el caso de que **high_out** sea menor que **low_out**, **imadjust** invierte la imagen de salida, como en un negativo fotográfico.

5.2. Cálculo del índice de Taguchi.

Como se ha mencionado previamente, el cálculo del ICM se basa exclusivamente en medir la distancia entre dos rectas paralelas, según se describe en la sección cuatro. A continuación se siguen los mismos pasos que los descritos en dicha sección.

5.2.1. Cálculo automático del borde inferior.

5.2.1.1. Recorte de la imagen.

Como se indicó previamente, se permite hacer zoom en la imagen mediante un recorte de la misma, que aparecerá en la ventana principal de la aplicación, para ello se utiliza la función **imcrop** (Código Anexo - punto 2). Mediante esta función es posible seleccionar el área rectangular que se quiera recortar, almacenándose en una variable global la imagen original en el caso de que quiera restaurar el recorte, posteriormente se muestra el recorte en la ventana correspondiente de la aplicación.

En la figura 6(a) podemos observar el rectángulo del recorte, la figura 6(b) es la imagen ya recortada.

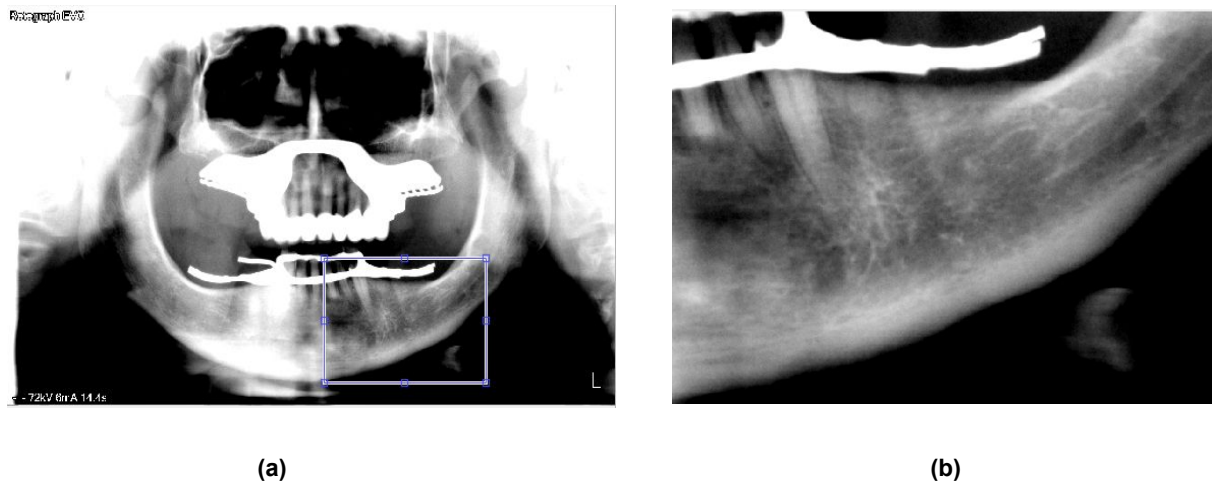


Figura 6. (a) Selección rectangular de la imagen; (b) Sección ampliada.

5.2.1.2. Binarización de la imagen.

A partir del recorte en la escala de grises realizada se obtiene la imagen binarizada mediante la función **im2bw** (Código Anexo - punto 3). Esta función reemplaza todos los píxeles en la imagen original con una luminancia mayor que un rango por el valor 1 (blanco) y todos los demás píxeles por el valor 0.

Recibe por parámetro la imagen original y un rango que se calcula mediante la función **graythresh** (Código Anexo - 4) que implementa el método de Otsu, obteniendo el umbral global correspondiente que se usa para convertir una imagen de intensidad en una imagen binaria. La figura 7(a) muestra el resultado de la imagen recortada, mostrándose en 7(b) la resultante de aplicar el proceso de binarización.

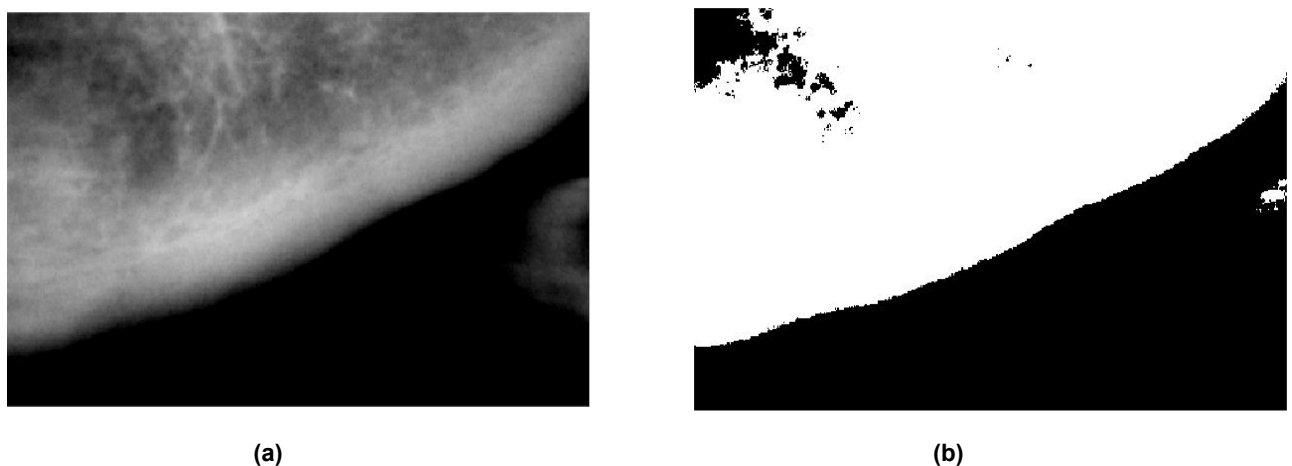


Figura 7. (a) Imagen original; (b) Imagen binarizada.

5.2.1.3. Limpieza de ruido.

Para limpiar la imagen de píxeles innecesarios se utiliza la función **bwareaopen** (Código Anexo - punto 5) que elimina todas las componentes conexas que tengan menos de **p** píxeles, generando otra imagen binaria depurada. En la figura 8(a) se muestra la imagen binarizada y en la figura 8(b) la binarizada sin ruido.



Figura 8. (a) Imagen binarizada original; (b) Imagen binarizada sin ruido .

La conectividad por defecto es 8 para dos dimensiones, 26 para tres dimensiones, y **conndef(ndims(BW), 'maximal')** de dimensiones superiores.

5.2.1.4. Detección de bordes.

Para detectar los bordes de la imagen binarizada se utiliza la función **edge** (Código Anexo - punto 6) y la aproximación a la derivada por el método de Sobel, devuelve todos los bordes conteniendo un número de píxeles mayor que el valor del rango (threshold) introducido por parámetro. La figura 9 muestra la imagen resultante después de aplicar la función de detección de bordes.

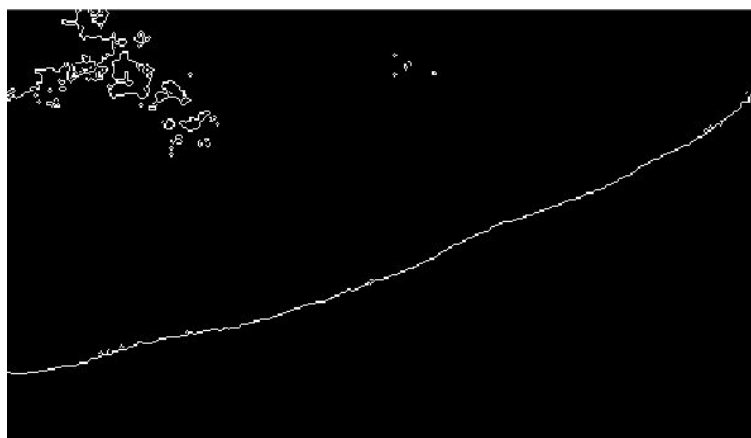


Figura 9. Detección de bordes mediante operador Sobel

5.2.2. Generación de recta ajustada al borde inferior.

5.2.2.1. Transformación a uint8.

Se transforma la imagen binarizada a uint8 con la función **im2uint8** (Código Anexo - punto 7), que permite manejar los tres canales RGB de la imagen con este formato propio.

5.2.2.2. Extracción de blancos.

Para generar las rectas debemos determinar las diferentes regiones que existen en la imagen. Mediante técnicas de color, se segmenta la imagen separando dos clases (blanco y negro). Para ello se utiliza la función llamada **extract_white_region** (Código Anexo - punto 8) determina las regiones en la imagen utilizando el sistema de color CIE-XYZ y devuelve una imagen con las regiones ya separadas.

Para determinar correctamente las regiones de la imagen refinamos el umbral de binarización con la función **rgb2gray** (Código Anexo - punto 9) que convierte la imagen RGB a escala de grises, eliminando la saturación y el matiz, pero conservando la luminosidad.

A posteriori extraemos y ampliamos los bordes para luego identificar las líneas que definen la región de la matriz, utilizamos la función **bwmorph** (Código Anexo - punto 10) aplica la operación morfológica sobre la imagen binaria un cierto número de veces que indiquemos por parámetro, en nuestro caso se aplica la operación de dilatación cinco veces, cuyo objetivo es ensanchar la línea hasta conseguir una cierta densidad de puntos que definan con mayor énfasis la línea recta que se quiere detectar.

5.2.2.3. Transformada de Hough.

Una vez obtenidas las regiones de la imagen el objetivo siguiente consiste en detectar todas las líneas rectas potenciales aplicando la transformada de Hough con la función **hough_Gonzalez** (Código Anexo - punto 11).

Tal y como se explica en Gonzalez y col (2004), el uso de la transformada de Hough para la localización y unión de líneas puede resultar una tarea ardua a la hora de encontrar un conjunto significativo de valores o picos en el acumulador, que corresponden a posibles rectas debido a dicho valor.

Debido al espacio cuantitativo de una imagen digital, dicha cuantificación en parámetros espaciales de la transformada de Hough y el hecho de que los bordes no son perfectamente rectos; los picos de la transformada de Hough tienden a situarse en más de una celda del espacio transformado de Hough. Por este motivo, la estrategia utilizada para resolver este problema es la siguiente:

1. Se encuentra la celda de Hough con el valor más alto y se guarda su localización.
2. Se asignan el valor cero a las celdas vecinas con respecto a la celda encontrada en el primer paso.
3. Se repite el procedimiento hasta que se encuentre un número deseado de picos significativos.

5.2.2.4. Unión de líneas similares.

Una vez identificados una serie de picos candidatos mediante la transformada de Hough, lo único que queda es determinar si hay segmentos de líneas asociadas con estos picos, además de conocer dónde empiezan y acaban. Para cada pico, el primer paso es localizar en la imagen todos los píxeles que no tengan valor 0 y que hayan contribuido a dicho pico, dichos píxeles se localizan con la función **hough_pixels**. (Gonzalez y col., 2004)

Estos píxeles deben agruparse en segmentos de líneas, procediendo posteriormente a la unión de líneas con parámetros (ρ, θ) similares se establece la longitud mínima de las mismas. El objetivo, es fusionar líneas con similar (ρ, θ) , pueden corresponder a segmentos de la misma línea que aparecen separados como consecuencia de la ruptura de la propia línea debido a los procesamientos previos o incluso a la baja calidad de las imágenes. Esto se consigue con la función **houghlines_Gonzalez** (Código Anexo - 13). **houghlines_Gonzalez** implementa la siguiente estrategia:

1. Rotar las localizaciones de los píxeles $90^\circ - \theta$ de tal forma que se sitúen aproximadamente junto a una vertical,
2. Ordenar las localizaciones de los píxeles por sus valores en x una vez rotados.
3. Usar la función **diff** para localizar huecos, se ignoran los huecos pequeños, esto permite unir segmentos de línea adyacentes que estén separados por estos huecos.
4. Devolver la información de los segmentos de línea que posean mayor longitud (en píxeles) que un determinado valor previamente establecido.

El resultado de los procedimientos descritos hasta aquí es la obtención automática de la primera línea tangencial al borde inferior de la mandíbula. La figura 10 muestra la recta generada mediante el proceso descrito con la transformada de Hough.

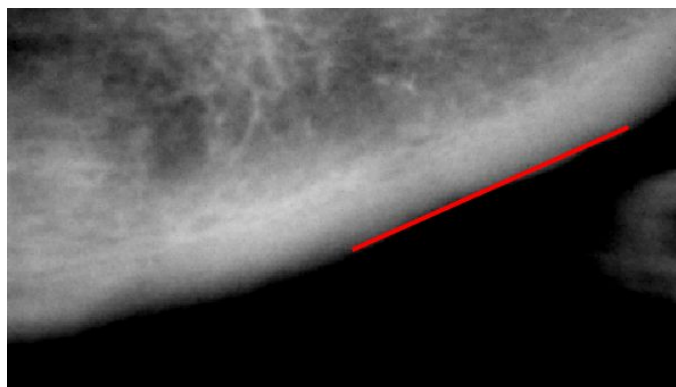


Figura 10. Recta autogenerada tangencial al borde inferior de la mandíbula.

5.2.3. Generación de la recta paralela.

Una vez obtenida la recta tangencial al borde inferior de la mandíbula se tiene que generar la recta tangencial al borde superior de la mandíbula y paralela a la anterior. El procedimiento implementado en este caso es el que se describe a continuación basado en los pasos explicados en la sección cuatro.

5.2.3.1. Cálculo de la pendiente.

Se obtienen dos puntos de la primera recta generada automáticamente y se calcula la pendiente de la recta (m) de la siguiente forma:

```
XR = [(xy(2,2)+position(1)),(xy(2,1)+position(2))];  
YR = [(xy(1,2)+position(1)),(xy(1,1)+position(2))];  
slope=(YR(2)-XR(2))/(YR(1)-XR(1));
```

5.2.3.2. Cálculo de coordenadas.

Para seleccionar el punto por el que pasará la segunda recta se utiliza la función **ginput**. Como una recta viene definida de forma básica mediante dos puntos, éstos se deben determinar en base a la ecuación de la recta.

```
b = Y-slope*X;  
Y4 = slope*XR(1)+b;  
Y5 = slope*YR(1)+b;
```

Siendo **b** el punto donde se produce la intersección de la recta con el eje Y; Siendo **Y4** e **Y5** los componentes del primer punto de la recta paralela e **XR(1)** e **YR(1)** los primeros componentes de la recta obtenida automáticamente.

Con esto se tienen ya los dos puntos necesarios para generar la recta paralela.

```
XV = [XR(1),YR(1)]  
YV = [Y4,Y5]
```

La figura 11 muestra la recta paralela resultante tras seleccionar el punto de paso definido por el usuario.

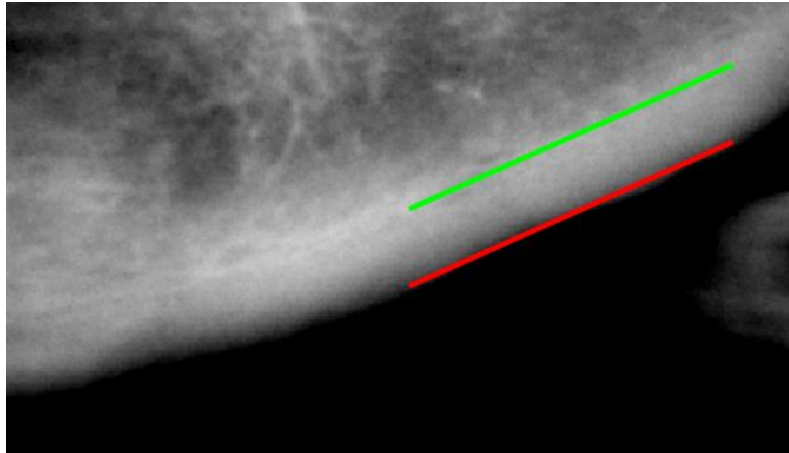


Figura 11. Recta paralela tangencial al borde superior de la mandíbula

5.2.4. Generación de la recta perpendicular.

Una vez calculadas las rectas tangenciales al borde inferior y superior de la mandíbula, ambas paralelas entre sí, sólo falta trazar una recta perpendicular para calcular la distancia entre los puntos de corte de éstas con la perpendicular.

5.2.4.1. Cálculo de la pendiente.

La pendiente de la recta perpendicular a otra recta dada, se obtiene calculando la inversa cambiada de signo respecto de la primera, tal y como se indica seguidamente.

```
slope2=-1/slope;
```

Siendo **slope** la pendiente de ambas rectas paralelas.

5.2.4.2. Cálculo de coordenadas.

En este momento se debe indicar el punto exacto, correspondiente al agujero mentoniano, por donde debe pasar la perpendicular a las dos rectas paralelas, ya trazadas, para ello se usa la función **ginput(1)**.

Después se debe calcular los dos puntos de la recta perpendicular para poder representarla.

```
X2= X - slope2 * (XR(2)-YR(2));  
Y2= Y + slope2 * (XR(1)-YR(1));  
XA=[X,X2];  
YA=[Y,Y2];
```

Siendo **XA** e **YA** los puntos de intersección de la recta perpendicular con la recta paralela

en la parte superior; y **XR** e **YR** los puntos de intersección también de la recta perpendicular con la recta tangencial a la parte inferior del mentón obtenida automáticamente.

La figura 12 muestra la línea perpendicular común a ambas rectas paralelas.

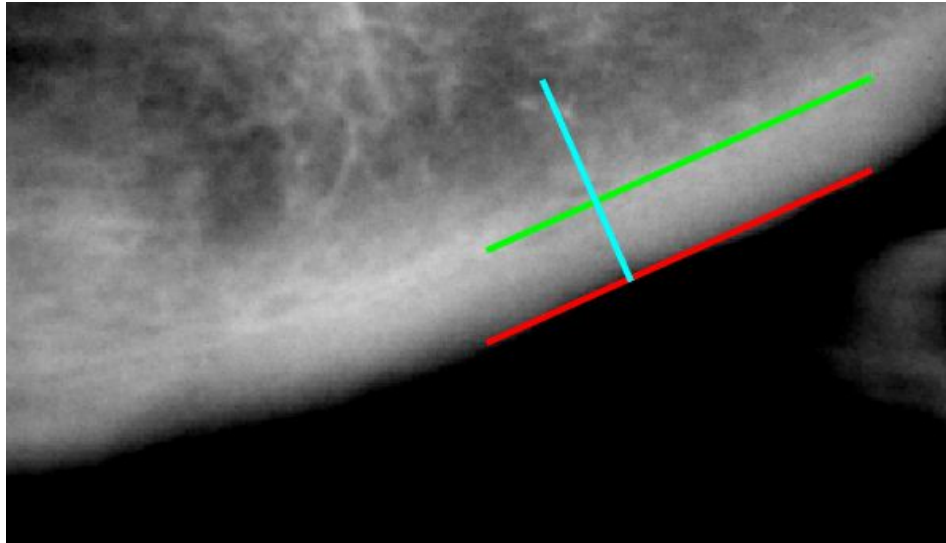


Figura 12. Recta perpendicular trazada desde el agujero mentoniano

5.2.5. Intersecciones entre rectas.

Una vez calculadas las tres rectas se calcula la distancia que las separa, obteniendo así el ICM.

5.2.5.1. Punto de intersección entre rectas.

El primer paso requerido para obtener el punto de intersección de dos rectas es identificar dos puntos pertenecientes a cada una de las rectas, lo que se consigue de la siguiente forma:

$$\begin{aligned} \mathbf{A} &= [\mathbf{XR}(1); \mathbf{XR}(2)]; \mathbf{B} = [\mathbf{YR}(1); \mathbf{YR}(2)]; \\ \mathbf{E} &= [\mathbf{XA}(1); \mathbf{YA}(1)]; \mathbf{F} = [\mathbf{XA}(2); \mathbf{YA}(2)]; \\ \mathbf{C} &= [\mathbf{YR}(2); \mathbf{YV}(2)]; \mathbf{D} = [\mathbf{YR}(1); \mathbf{YV}(1)]; \end{aligned}$$

Siendo **A** y **B** los dos puntos de la recta tangencial al borde inferior de la mandíbula; **C** y **D** los dos puntos de la recta tangencial al borde superior de la mandíbula; **E** y **F** los dos puntos de la recta perpendicular a las dos anteriores.

Con estos puntos identificados, mediante la función **lineIntersection** (Código Anexo - punto 14), se calcula el punto de intersección de la siguiente forma: $X = B-F$; $M = [(B-A), (E-F)]$; $Z = MX$; $\lambda = Z(1)$; $\gamma = Z(2)$;

5.2.5.2. Distancia entre los puntos generados (Índice Taguchi).

Llegados a este punto, sólo falta medir la distancia entre los dos puntos de intersección de la perpendicular con las dos rectas paralelas.. Para ello se calcula la distancia euclídea entre dos puntos, como sigue:

```
Dist = sqrt((E1(1)-E2(1))^2+(E1(2)-E2(2))^2);
```

Siendo **E1** el punto de corte entre la recta perpendicular y la recta tangencial al borde inferior de la mandíbula (línea roja en la figura 13); **E2** el punto de corte entre la recta perpendicular y la recta paralela al borde superior de la mandíbula (línea verde en la figura 13).

Precisamente, la figura 13 muestra la intersección entre la recta tangencial al borde inferior de la mandíbula y la paralela a la misma. En la parte inferior de la imagen se muestra el valor de la distancia entre las rectas, y por tanto el ICM.

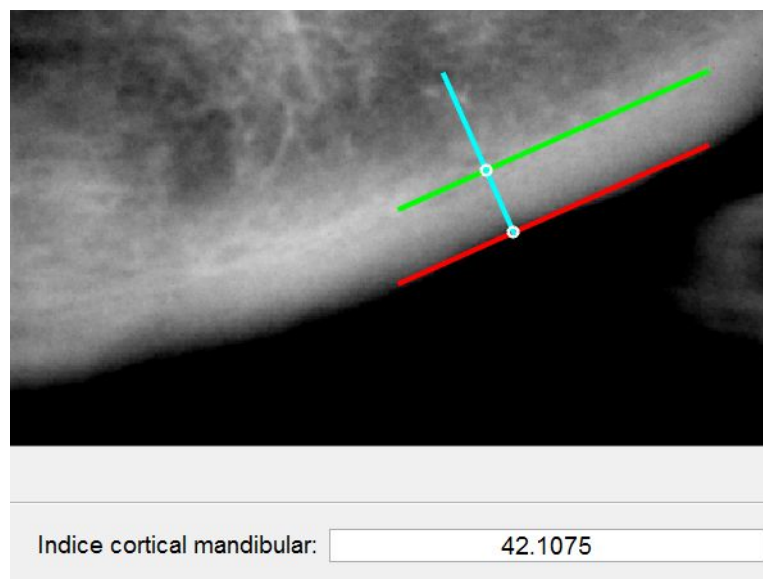


Figura 13. Intersección entre rectas e ICM calculado

5.3. Descripción de texturas

Tal y como se ha descrito en la sección seis, el análisis de texturas se realiza sobre un recorte de la imagen original disponible, lo cual se lleva a cabo utilizando la función **imcrop** anteriormente explicada.

A continuación se crea un contenedor **I1** de las mismas dimensiones **MxN** que el recorte para almacenar los valores de textura. Los valores de este contenedor son cuyos valores iniciales son cero. Para ello se utiliza la función de matlab **zeros(M,N)**, siendo **M** y **N** las dimensiones deseadas. Se procede entonces al cálculo de la desviación típica sobre cada píxel de la región de interés seleccionada en una región de vecindad de dimensión 3x3, identificada como **v**. Este cálculo se realiza mediante la función **std2(v)**.

Se utiliza la función de MatLab **imagesc(I1)** la cual muestra los datos en la matriz **I1** como una imagen que utiliza una gama completa de colores según el mapa de colores. Cada elemento de **I1** se muestra con un color determinado en función de su valor.

La imagen resultante permite realizar el análisis sobre los contrastes de intensidad en la textura ósea del paciente. La figura 14 muestra los diferentes contrastes de texturas, según la escala de colores situada a la derecha de la imagen resultante.

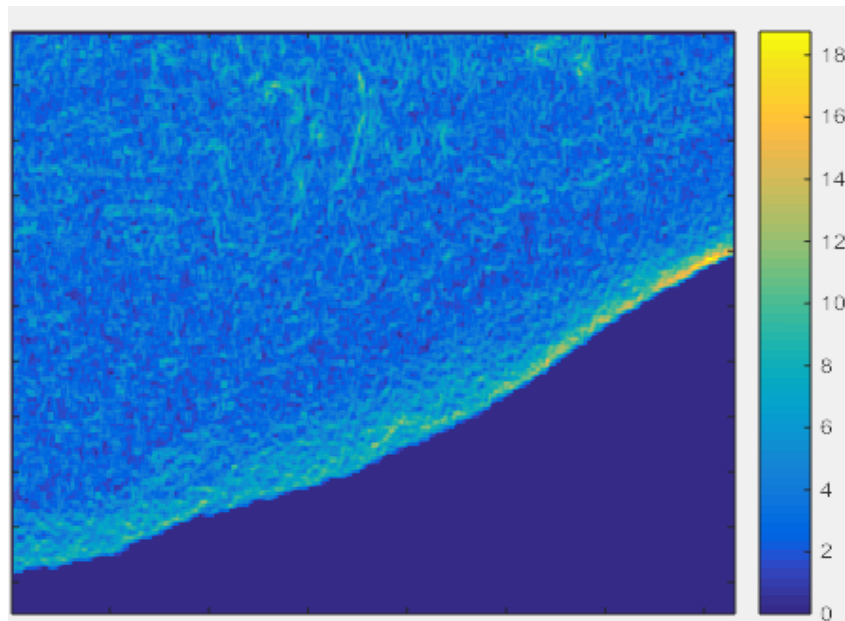


Figura 14. Imagen resultante del cálculo de texturas

6. Conclusiones y trabajo futuro.

Llegados a este punto, se está en condiciones de afirmar que se ha cumplido con el objetivo principal del proyecto. Se ha desarrollado una aplicación capaz de calcular correctamente el ICM y el análisis de la descripción de las texturas de una radiografía. Se ha creado una estructura que permite la implementación de distintos módulos de una manera rápida y sencilla gracias al uso de patrones de diseño.

El código generado a lo largo de la vida del proyecto es altamente reutilizable y flexible, lo que facilita la creación de distintas vías y oportunidades de desarrollo para continuar con el trabajo realizado en futuros desarrollos. El uso de la herramienta de desarrollo Matlab ha resultado un gran acierto pues nos ha permitido una gran modulación del código posibilitando la adición de nuevas funcionalidades.

Hubiese sido deseable lograr una automatización completa del cálculo del ICM para facilitar y agilizar a los profesionales la detección de la osteoporosis del paciente. Esto no ha sido posible debido a la baja calidad de las radiografías panorámicas y de la dificultad de detectar de forma automática el agujero mentoniano necesario para el cálculo del ICM, en este sentido se requiere un esfuerzo adicional en el futuro.

La resolución del cálculo del ICM ha dado buenos resultados, devolviendo aproximaciones bastante acertadas a medida que se acotaban los umbrales de intensidad. Otra posibilidad de cambio o ampliación futura es incluir la funcionalidad de la clasificación de grises mediante métodos específicos basados en técnicas de aprendizaje automático. Esta funcionalidad permitiría realzar diferentes tonalidades de grises en la radiografía, facilitando así la detección de posibles infecciones sobre la estructura ósea del paciente.

En este proyecto se han combinado diferentes tecnologías, algunas de ellas ciertamente novedosas. En este sentido se han utilizado técnicas específicas de tratamiento de imágenes en el marco de una aplicación de desarrollo software, lo que ha permitido verificar la perfecta sintonía entre ambas tecnologías.

La implementación de patrones, mantener los factores de calidad del software y los conocimientos adquiridos sobre percepción computacional han sido piezas claves para poder desarrollar este proyecto.

6. Conclusions and future work.

At this point, we may say that the main objectives of the project have been accomplished. We have developed a program capable of calculating the mandibular cortical index and perform the analysis of texture description correctly. We have also created a structure which allows the implementation of different modules in an easy and quick way thanks to the use of design patterns.

The code generated through the lifetime of the project is highly reusable and flexible, allowing the easy creation of different ways of development in order to continue with the project in future works. The use of Matlab has been a great success, because it allows a great code modulation in order to add new functionalities.

It would have been desirable to achieve a complete automatization of the ICM calculation, so dental professional found easier and quicker to detect signs of osteoporosis on the patients. This was not possible because of the low quality of the panoramic radiographies and the difficulty to detect the mental foramen hole which is needed to calculate the ICM.

The resolution of the ICM let us achieve good results, showing accurate approximations while we improved the intensity threshold. Another future work is the inclusion of gray classification. This functionality allows to detect different gray tones on the radiography, this makes the detection of possible infections on the bone structure easy.

In this project we have combined several technologies, some of them certainly new. Several specific techniques of image processing has been used in order to develop a software application, this has allowed a perfect harmony to exist between both technologies.

The implementation of patterns, the maintaining of software quality factors and the knowledge acquired of computational perception have been key for developing this project parts.

Bibliografía.

1. Abhyankar, S.S., Shriram, R. (2013). Orthopantogram based Evaluation of Osteoporosis by Studying Jaw Bone Changes. International Journal of Computer Applications, 66(17), 36-39.
2. Hastar, E., Yilmaz, H.H., Orhan, H. (2011). Evaluation of mental index, mandibular cortical index and panoramic mandibular index on dental panoramic radiographs in the elderly. Eur J. Dent., 5, 60-7.
3. Gonzalez R. C., Woods R.E., Eddins, S.L. (2004). Digital Image Processing using Matlab, Prentice Hall, Upper Saddle River, NJ.
4. Hough, P.V.C. (1962). Method and means for recognizing complex patterns. U. S. Patent 3, 069 654, December 18, 1962.
5. Kavitha, M.S., An, S.Y., An, C.H. Huh, K.H., Yi, W.J. Heo, M.S., Heo, M.S., Lee, S.S., Choi, S.C. (2014). Texture analysis of mandibular cortical bone on digital dental panoramic radiographs for the diagnosis of osteoporosis in Korean women. Oral Surgery, Oral Medicine Oral Pathology Oral Radiology, 119, 3, 346-356.
Disponible: <http://www.sciencedirect.com/science/article/pii/S2212440314013935> (accedido junio 2017).
6. Marandi, S., Bagherpour, A., Imanimoghaddam, M., Hatef, M., Haghigh, A. (2010). Panoramic-based mandibular indices and bone mineral density of femoral neck and lumbar vertebrae in women. Journal Dent. https://openi.nlm.nih.gov/detailedresult.php?img=PMC3184742_jod-7-098-g002&req=4
7. Martínez, T. (2017) La Transformada de Hough. Detección de Líneas y Círculos.

Disponible:

<https://porprofesionalmic.files.wordpress.com/2015/09/investigacion-documental-transformada-hough.pdf> (accedido junio 2017).

8. Matlab (2017) The Matworks <https://es.mathworks.com/> (accedido junio 2017).
9. Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Trans. Systems Man Cybernetics, SMC-9, 62-66.
10. Pajares, G. (2007); Visión por computador: Imágenes digitales y aplicaciones, RA-MA, Madrid
11. Sobel, I. (2015) An Isotropic 3x3 Image Gradient Operator. DOI: 10.13140/RG.2.1.1912.4965 Disponible: https://www.researchgate.net/publication/281104656_An_Isotropic_3x3_Image_Gradient_Operator (accedido mayo 2017).
12. Taguchi A. (2010) Triage screening for osteoporosis in dental clinics using panoramic radiographs. Oral Dis.; 16(4), 316-27.

Anexos: Manual de Usuario y Código.

El presente apéndice describe dos aspectos importantes relativos al desarrollo de la aplicación. Por un lado el Manual de Usuario, que proporciona los detalles de uso de la aplicación y por otro el código necesario para el ajuste de la línea recta sobre el borde inferior de la mandíbula mediante la transformada de Hough.

Anexo A. Manual de usuario.

En este apartado se explica cómo usar la aplicación paso a paso por cada funcionalidad.

La figura 15 muestra el panel de inicio de la aplicación, como se puede observar solamente se puede proceder a la carga de la imagen, y una vez realizada se habilitan el resto de los controles del usuario.

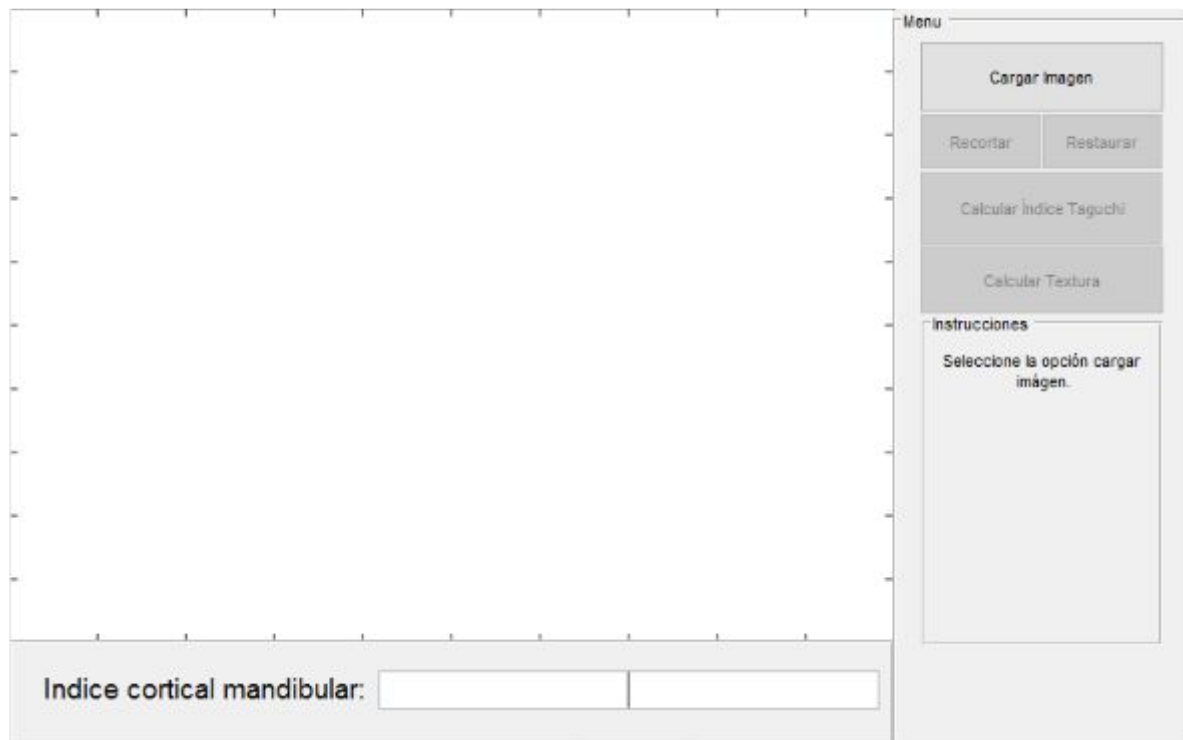


Figura 15. Interfaz de la aplicación al iniciar

La primera operación consiste en cargar la imagen a tratar, para ello se utilizará el botón **“Cargar Imagen”**.

Con tal propósito se ha habilitado una carga estándar de archivos, que se seleccionan de forma individual para visualización de la imagen bajo tratamiento. El proceso de selección se muestra en la imagen 16.

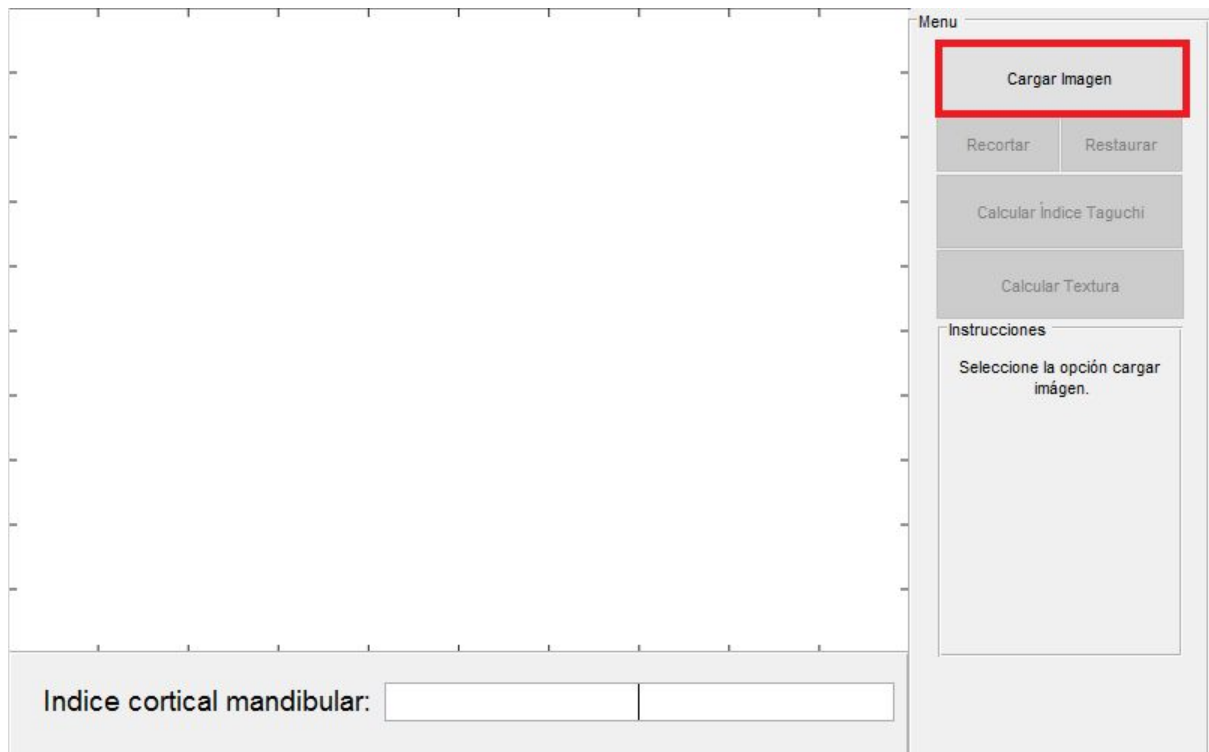


Figura 16. Botón de la GUI que permite la carga de archivos

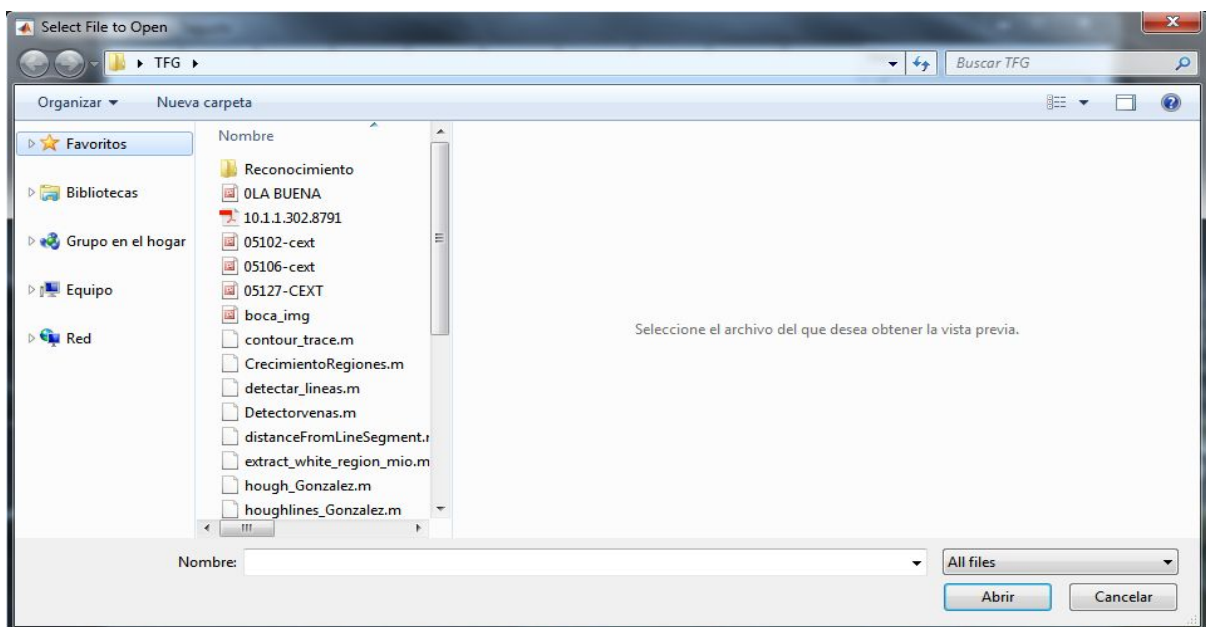


Figura 17. Carga de archivos en la aplicación

A1) Visualización radiografía

Una vez cargada la imagen visualizada, ésta se sitúa en disposición de proceder al tratamiento sobre la misma, tal y como se ha descrito a lo largo de la memoria. Tras esta operación se habilitan los botones correspondientes al recorte de la imagen, así como el de calcular el ICM y las texturas, tal y como se visualiza en la figura 18

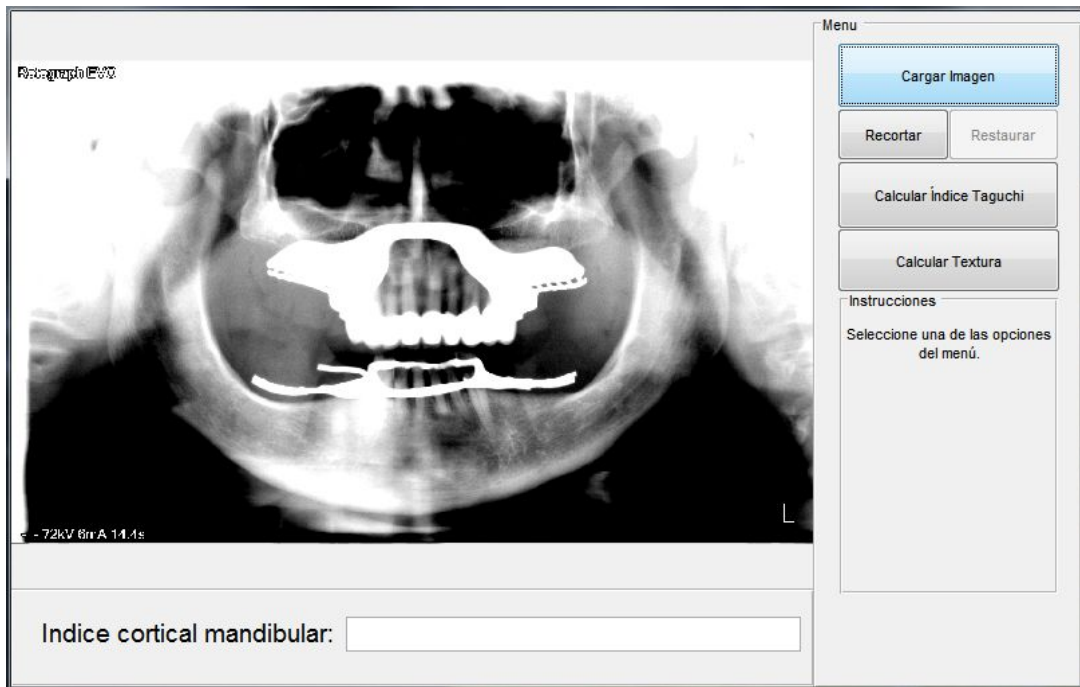


Figura 18. Visualización de la radiografía

A2) Opción recortar

Si el usuario desea analizar una región concreta de la imagen con mayor detalle se permite dicha operación mediante la función de **“Recortar”**. Se podrá hacer el recorte posicionando y pulsando el cursor del ratón sobre una zona de la imagen y arrastrándolo para generar un rectángulo que define el recorte. Si se desea volver a visualizar la imagen completa es necesario seleccionar el botón **“Restaurar”**. Estas operaciones se muestran en la figura 19.

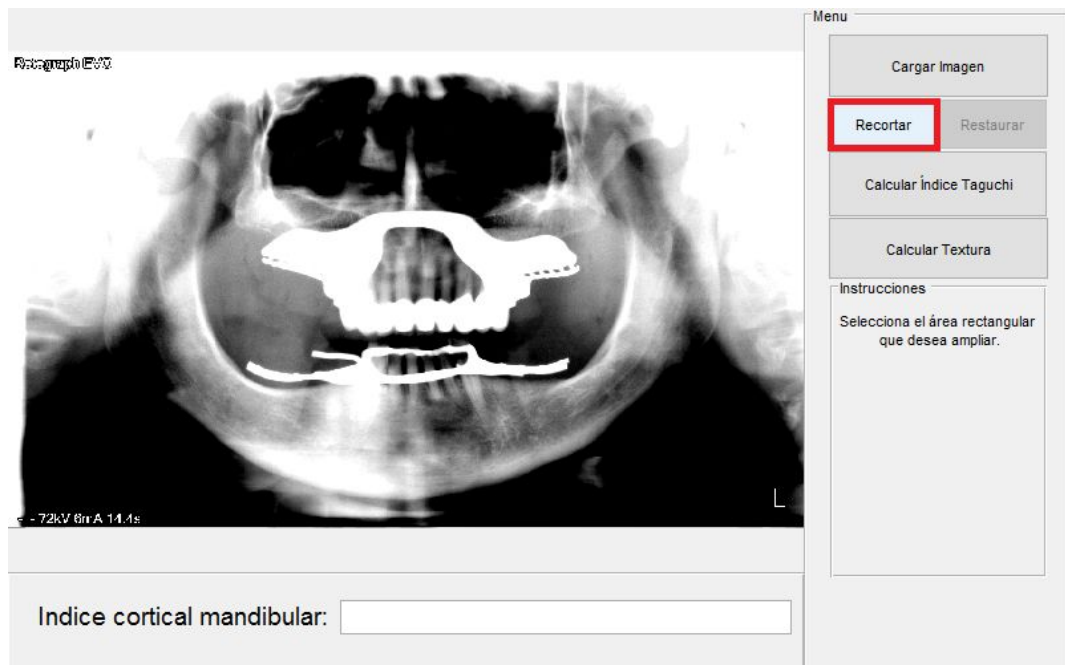


Figura 19. Botón de la GUI que permite el recorte de la imagen.

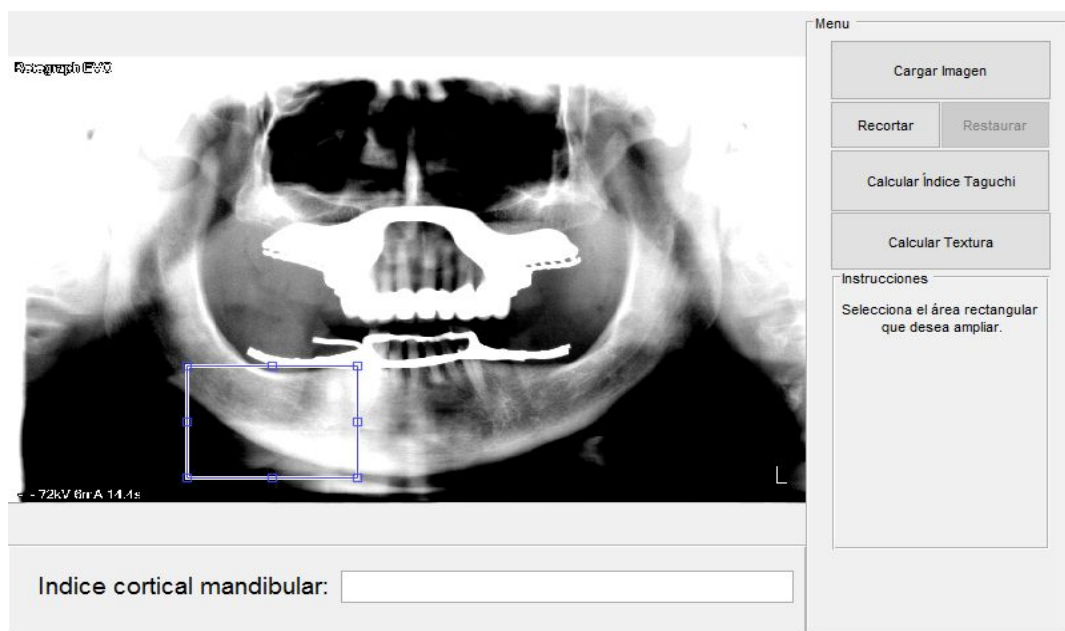


Figura 20. Recuadro para seleccionar la zona de recorte.

A3) Cálculo del ICM

Al pulsar sobre la opción calcular “**Índice Taguchi**”, en primer lugar, se proporciona la opción de aplicar un recorte sobre la imagen original del mismo modo que el descrito previamente, tras lo cual se desencadena el proceso de ajuste automático de la recta tangencial inferior a la mandíbula, tal y como muestra la línea roja en la imagen de la figura 21.

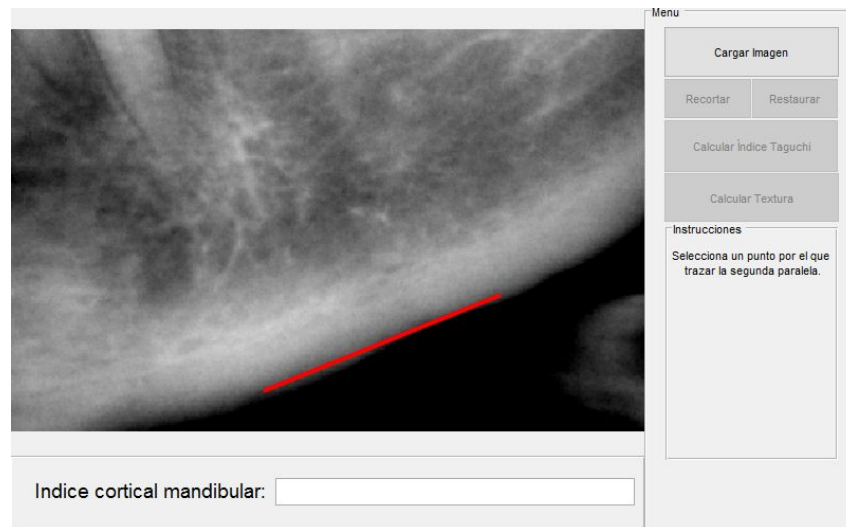


Figura 21. Recta auto detectada del borde inferior de la mandíbula.

Para generar la segunda recta, paralela a la anterior, en la parte superior del borde de la mandíbula, se indica al usuario la necesidad de seleccionar un punto sobre el borde superior, para lo cual se pulsa con el botón del ratón en el punto seleccionado. Tras esta operación se dibuja la línea deseada, tal y como se muestra en la figura 22, representada por la línea verde.

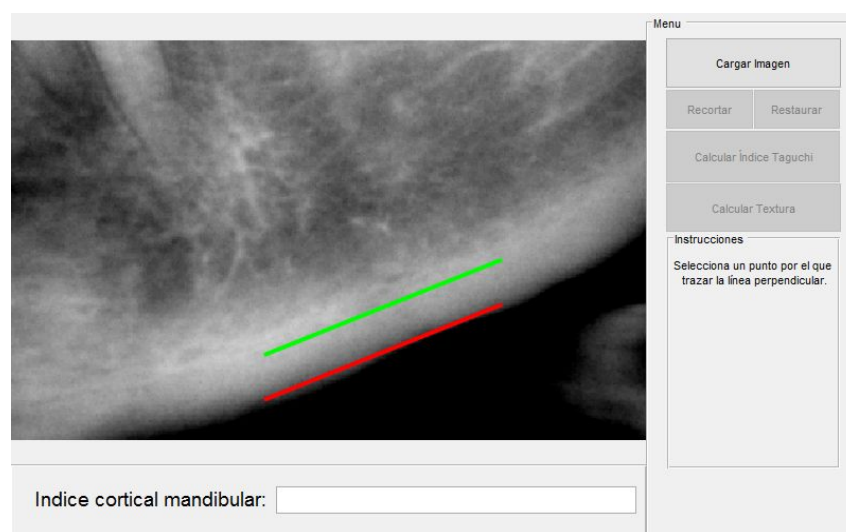


Figura 22. Recta paralela a la recta autogenerada.

Por último, se debe seleccionar un punto para trazar la recta perpendicular, línea azul en la figura 23, a ambas rectas paralelas previamente obtenidas. Para ello es necesario pulsar con el botón del ratón sobre el agujero mentoniano. Sobre esta línea se visualizarán dos marcas que representan los puntos de corte con las rectas paralelas calculadas anteriormente, tras lo cual aparece el valor del ICM en la parte inferior del panel.

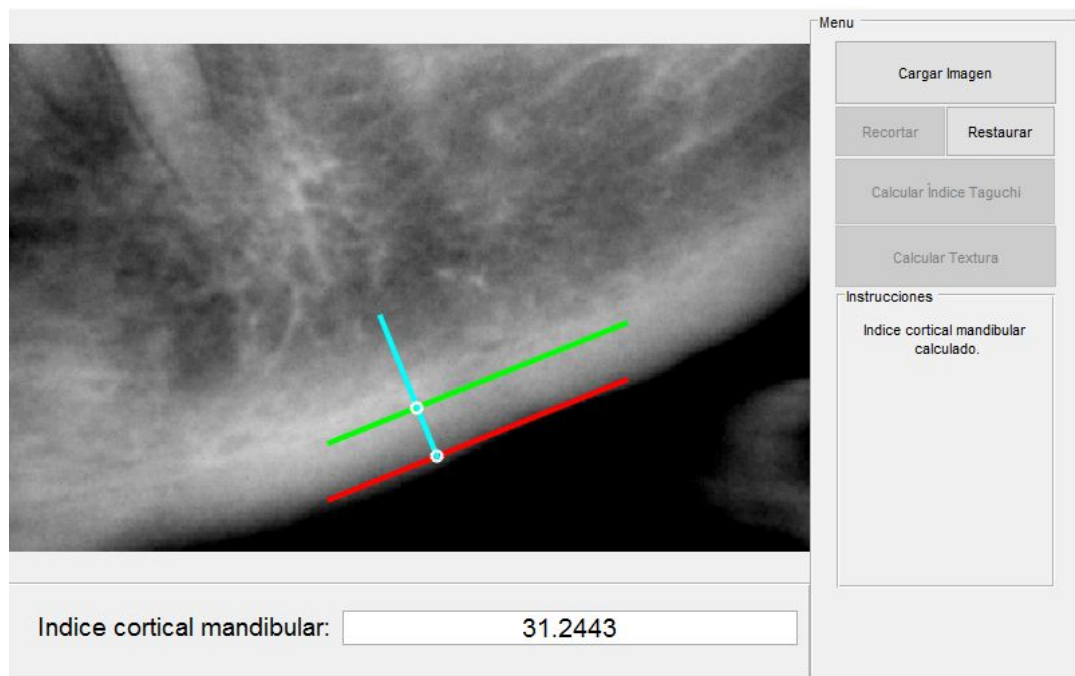


Figura 23. Recta perpendicular a ambas rectas y el cálculo del ICM.

Es necesario remarcar, que se podrá volver a la imagen original una vez hecho el recorte de zoom y/o al finalizar el proceso del cálculo del ICM pulsando sobre el botón **“Restaurar”**, que eliminará cualquier cambio hecho, mostrando la imagen original de nuevo en la ventana principal.

A4) Descripción de Texturas

Para aplicar esta función basta con pulsar el botón “Calcular Textura”.

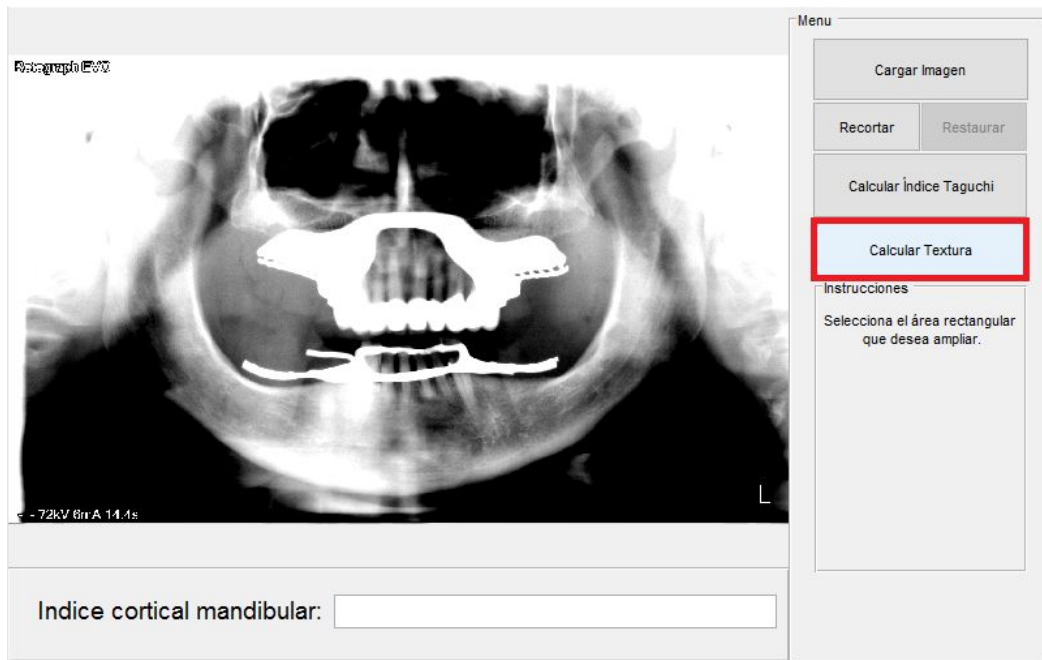


Figura 24. Botón de la GUI que permite el cálculo de la textura.

Tendremos que señalar la zona a analizar señalando con un rectángulo. Posteriormente se mostrará la zona recortada y la visualización de las diferentes texturas, figura 25.

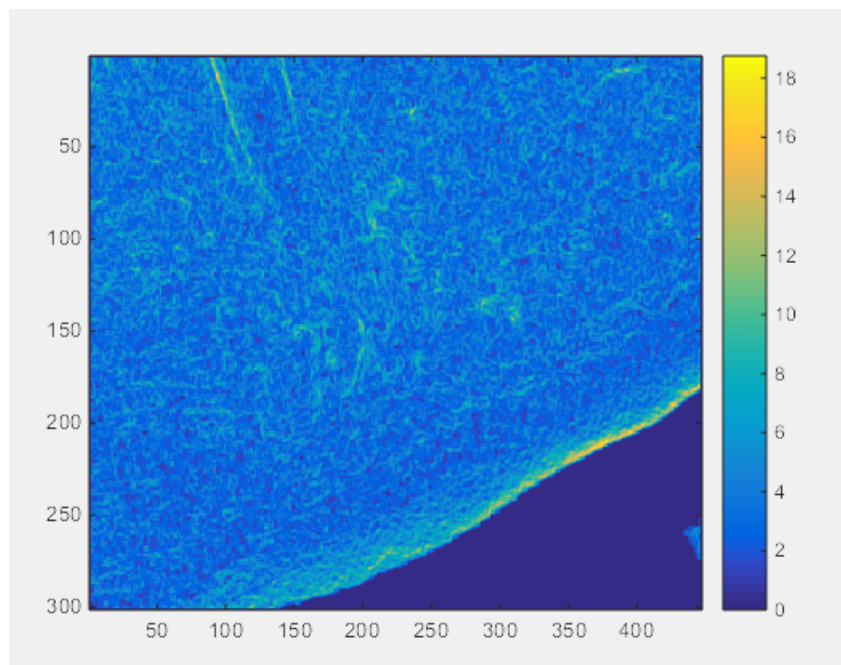


Figura 25. Gráfico de intensidad de texturas

Anexo B. Código

La siguiente secuencia de código muestra la implementación realizada para ajustar la línea recta sobre la parte inferior de la mandíbula mediante la transformada de Hough.

```
1) J = imadjust(I,[low_in; high_in],[low_out; high_out]);
2) J = imcrop(I,rect);
3) BW = im2bw(RGB, level);
4) level = graythresh(I);
5) BW2 = bwareaopen(BW,P);
6) BW = edge(I,'Sobel',threshold)
7) I2 = im2uint8(I);
8) [f] = extract_white_region_mio(I);
9) f = rgb2gray(image);
10) y2 = bwmorph(g,operation,n);
11) [H,Theta,Rho] = hough_Gonzalez(f,theta,rho);
12) [filas,columnas,Hnew] = houghpeaks_Gonzalez(H,numPicos,Trheshold);
13) lineas =
    houghlines_Gonzalez(f,Theta,Rho,filas,columnas,FillGap,Minima_longitud);
14) [E1, lambda, gamma, isConvex] = lineIntersection(A,B,E,F);
```